

```

LLL      000000000    AAAAAAAAAA    DDDDDDDDDDDDD    SSSSSSSSSSSSS    SSSSSSSSSSSSS
LLL      000000000    AAAAAAAAAA    DDDDDDDDDDDDD    SSSSSSSSSSSSS    SSSSSSSSSSSSS
LLL      000000000    AAAAAAAAAA    DDDDDDDDDDDDD    SSSSSSSSSSSSS    SSSSSSSSSSSSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSC      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAAAAAAAAAAAAAA    DDD      DDD    SSS      SSS
LLL      000      000    AAAAAAAAAAAAAAA    DDD      DDD    SSS      SSS
LLL      000      000    AAAAAAAAAAAAAAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLL      000      000    AAA      AAA    DDD      DDD    SSS      SSS
LLLLLLLLLLLLLLLLLL    000000000    AAA      AAA    DDDDDDDDDDDDD    SSSSSSSSSSSSS    SSSSSSSSSSSSS
LLLLL.LLLLLLLLLLLL    000000000    AAA      AAA    DDDDDDDDDDDDD    SSSSSSSSSSSSS    SSSSSSSSSSSSS
LLLLLLLLLLLLLLLLLL    000000000    AAA      AAA    DDDDDDDDDDDDD    SSSSSSSSSSSSS    SSSSSSSSSSSSS

```

[illegible]

```
1 0001 0 MODULE SYSACLSRV (  
2 0002 0     LANGUAGE (BLISS32),  
3 0003 0     IDENT = 'V04-000',  
4 0004 0     ADDRESSING_MODE (EXTERNAL = GENERAL,  
5 0005 0     NONEXTERNAL = LONG_RELATIVE)  
6 0006 0 ) =  
7 0007 1 BEGIN  
8 0008 1  
9 0009 1 *****  
10 0010 1 *  
11 0011 1 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
12 0012 1 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
13 0013 1 *  ALL RIGHTS RESERVED.  
14 0014 1 *  
15 0015 1 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
16 0016 1 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
17 0017 1 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
18 0018 1 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
19 0019 1 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
20 0020 1 *  TRANSFERRED.  
21 0021 1 *  
22 0022 1 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
23 0023 1 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
24 0024 1 *  CORPORATION.  
25 0025 1 *  
26 0026 1 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
27 0027 1 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
28 0028 1 *  
29 0029 1 *  
30 0030 1 *****  
31 0031 1  
32 0032 1 ++  
33 0033 1  
34 0034 1 FACILITY:      Miscellaneous system services  
35 0035 1  
36 0036 1 ABSTRACT:  
37 0037 1  
38 0038 1     This module contains the routines necessary to convert an Access  
39 0039 1     Control Entry from a user (text) format to an internal (binary)  
40 0040 1     format and back again.  
41 0041 1  
42 0042 1 ENVIRONMENT:  
43 0043 1  
44 0044 1     VAX/VMS operating system, non-privileged system services.  
45 0045 1  
46 0046 1 --  
47 0047 1  
48 0048 1  
49 0049 1 AUTHOR:      L. Mark Pilant      CREATION DATE: 30-Sep-1982 11:00  
50 0050 1  
51 0051 1 MODIFIED BY:  
52 0052 1  
53 0053 1     V03-021 LMP0299      L. Mark Pilant,      8-Aug-1984 12:31  
54 0054 1     Require SYSPRV to add an ACL to a device if it is unowned  
55 0055 1     and there is no ACL already present on the device.  
56 0056 1  
57 0057 1     V03-020 LMP0275      L. Mark Pilant,      11-Jul-1984 10:11
```


| | | | |
|-----|------|---|--|
| 58 | 0058 | 1 | |
| 59 | 0059 | 1 | |
| 60 | 0060 | 1 | |
| 61 | 0061 | 1 | |
| 62 | 0062 | 1 | |
| 63 | 0063 | 1 | |
| 64 | 0064 | 1 | |
| 65 | 0065 | 1 | |
| 66 | 0066 | 1 | |
| 67 | 0067 | 1 | |
| 68 | 0068 | 1 | |
| 69 | 0069 | 1 | |
| 70 | 0070 | 1 | |
| 71 | 0071 | 1 | |
| 72 | 0072 | 1 | |
| 73 | 0073 | 1 | |
| 74 | 0074 | 1 | |
| 75 | 0075 | 1 | |
| 76 | 0076 | 1 | |
| 77 | 0077 | 1 | |
| 78 | 0078 | 1 | |
| 79 | 0079 | 1 | |
| 80 | 0080 | 1 | |
| 81 | 0081 | 1 | |
| 82 | 0082 | 1 | |
| 83 | 0083 | 1 | |
| 84 | 0084 | 1 | |
| 85 | 0085 | 1 | |
| 86 | 0086 | 1 | |
| 87 | 0087 | 1 | |
| 88 | 0088 | 1 | |
| 89 | 0089 | 1 | |
| 90 | 0090 | 1 | |
| 91 | 0091 | 1 | |
| 92 | 0092 | 1 | |
| 93 | 0093 | 1 | |
| 94 | 0094 | 1 | |
| 95 | 0095 | 1 | |
| 96 | 0096 | 1 | |
| 97 | 0097 | 1 | |
| 98 | 0098 | 1 | |
| 99 | 0099 | 1 | |
| 100 | 0100 | 1 | |
| 101 | 0101 | 1 | |
| 102 | 0102 | 1 | |
| 103 | 0103 | 1 | |
| 104 | 0104 | 1 | |
| 105 | 0105 | 1 | |
| 106 | 0106 | 1 | |
| 107 | 0107 | 1 | |
| 108 | 0108 | 1 | |
| 109 | 0109 | 1 | |
| 110 | 0110 | 1 | |
| 111 | 0111 | 1 | |
| 112 | 0112 | 1 | |
| 113 | 0113 | 1 | |
| 114 | 0114 | 1 | |

If the ACL defined in the ORB is not a queue, initialize it. (This is to optimize the path through EXESCHKPRO_INT.)

V03-019 LMP0271 L. Mark Pilant, 29-Jun-1984 13:18
Add support for ORBSV_NOACL. (This is used by shared memory mailboxes to disallow ACLs.)

V03-018 LMP0261 L. Mark Pilant, 26-Jun-1984 11:25
Fix a bug that caused the success status to be dropped from the formatted AUDIT type ACE.

V03-017 LMP0248 L. Mark Pilant, 3-May-1984 14:30
Include the FID in the lock resource name for files, so that unique lock names will be generated. Also, do the unlock regardless of the status if indicated by the item list.

V03-016 LMP0243 L. Mark Pilant, 1-May-1984 8:45
Fix the dispatching so that if an error occurs, processing of the item list is not aborted. However, only items that do not alter the ACL are processed.

V03-015 ACG0417 Andrew C. Goldstein, 18-Apr-1984 11:50
Fix probing and lock access mode problems

V03-014 ACG0415 Andrew C. Goldstein, 30-Mar-1984 17:49
Add ACL mutex handling to CHANGE_ACL; add options parsing to default protection ACE; break out ACL processing subroutines into separate common module; mask success and fail flags in audit display; remove IOSB parameter in \$CHANGE_ACL; defer unlocking of file ACL until end of operation; find UCB ACL listhead in ORB instead of UCB.

V03-013 LMP0222 L. Mark Pilant, 27-Mar-1984 14:22
Change the tie off symbols, necessary because of linking with XQP module ALLOCB, from locations to offsets.

V03-012 LMP0213 L. Mark Pilant, 13-Mar-1984 9:53
Add support for exclusively locking and unlocking the object for ACL modifications.

V03-011 LMP0185 L. Mark Pilant, 2-Feb-1984 16:25
Add support for device ACLs. Also, improve the error handling considerably through judicious use of PROBing.

V03-010 LMP0179 L. Mark Pilant, 8-Dec-1983 15:46
Add a new routine SYSSCHANGE_ACL, for changing the ACL associated with an object.

V03-009 LMP0174 L. Mark Pilant, 2-Dec-1983 9:58
Add support for RMS journal-ID ACEs.

V03-008 LMP0170 L. Mark Pilant, 1-Dec-1983 16:43
Add support for the NOPROPAGATE flag.

V03-007 LMP0152 L. Mark Pilant, 12-Sep-1983 15:13
Make SECURITY the journal name for AUDIT and ALARM ACEs.


```

: 115      0115 1 |
: 116      0116 1 |
: 117      0117 1 |
: 118      0118 1 |
: 119      0119 1 |
: 120      0120 1 |
: 121      0121 1 |
: 122      0122 1 |
: 123      0123 1 |
: 124      0124 1 |
: 125      0125 1 |
: 126      0126 1 |
: 127      0127 1 |
: 128      0128 1 |
: 129      0129 1 |
: 130      0130 1 |
: 131      0131 1 |
: 132      0132 1 |
: 133      0133 1 |
: 134      0134 1 |
: 135      0135 1 |
: 136      0136 1 |
: 137      0137 1 |

V03-006 LMP0140      L. Mark Pilant,      23-Aug-1983  20:21
          Add support for alphanumeric UICs.

V03-005 LMP0135      L. Mark Pilant,      8-Aug-1983  11:03
          Change the parsing and formatting of directory default
          ACEs slightly.

V03-004 LMP0123      L. Mark Pilant,      22-Jun-1983  10:36
          Change the name of the FLAGS field to OPTIONS.

V03-003 LMP0122      L. Mark Pilant,      20-Jun-1983   9:14
          Add support for a directory default protection ACE.

V03-002 LMP0114      L. Mark Pilant,      11-May-1983  10:42
          Add support for an access bitmask name table.

V03-001 LMP0103      L. Mark Pilant,      24-Apr-1983  19:14
          Add support for HIDDEN and PROTECTED ACEs.

: 134      0134 1 |**
: 135      0135 1 |
: 136      0136 1 |
: 137      0137 1 |
          LIBRARY 'SYSS$LIBRARY:LIB.L32';
          LIBRARY 'SYSS$LIBRARY:TPAMAC.L32';
```

```
139 0138 1 ! Declare necessary builtin functions.
140 0139 1
141 0140 1 BUILTIN
142 0141 1     TESTBITSC,
143 0142 1     INSQUE,
144 0143 1     MOVPSL,
145 0144 1     MTPR,
146 0145 1     PROBER,
147 0146 1     PROBEW,
148 0147 1     REMQUE;
149 0148 1
150 0149 1 LINKAGE
151 0150 1     L_PROBE      = JSB (REGISTER = 3, REGISTER = 1, REGISTER = 0)
152 0151 1               : NOPRESERVE (2)
153 0152 1               : NOTUSED (4, 5, 6, 7, 8, 9, 10, 11),
154 0153 1
155 0154 1     L_VERIFY   = JSB (REGISTER = 0; REGISTER = 1)
156 0155 1               : NOPRESERVE (2, 3)
157 0156 1               : NOTUSED (4, 5, 6, 7, 8, 9, 10, 11),
158 0157 1
159 0158 1     L_MUTEX    = JSB (REGISTER = 0, REGISTER = 4)
160 0159 1               : NOTUSED (5, 6, 7, 8, 9, 10, 11);
161 0160 1
162 0161 1 FORWARD ROUTINE
163 0162 1     SYSSPARSE_ACL,      ! Convert ACE to binary
164 0163 1     SYSSFORMAT_ACL, ! Convert ACE to text
165 0164 1     SYSSCHANGE_ACL, ! Change an object's ACL
166 0165 1     GET_PARENT_LOCK, ! Take out parent for ACL locks
167 0166 1
168 0167 1 ! TPARSE action routine
169 0168 1
170 0169 1     SET_ID,              ! Save a converted identifier
171 0170 1     SET_ACCESS_BIT, ! Set desired access bit by name
172 0171 1
173 0172 1 ! ACL queue head locating routines.
174 0173 1
175 0174 1     GET_UCB_ACL,      ! For UCBs
176 0175 1     GET_JBC_ACL,  ! For Job controller queue
177 0176 1     GET_CEB_ACL,   ! For CEBs
178 0177 1     GET_LNT_ACL,  ! For logical name tables
179 0178 1     GET_PCB_ACL,   ! For processes
180 0179 1     GET_GBL_ACL,   ! For global sections
181 0180 1
182 0181 1 ! ACL action routines.
183 0182 1
184 0183 1     ACL_DISPATCH,      ! Main ACL function dispatcher
185 0184 1     RUNDOWN_CHANGE_ACL; ! Clean up $CHANGE_ACL context
186 0185 1
187 0186 1 EXTERNAL ROUTINE
188 0187 1     ACL_ADDENTRY,      ! Add an ACE
189 0188 1     ACL_DELENTY,     ! Delete an ACE
190 0189 1     ACL_MODENTRY,   ! Modify an ACE
191 0190 1     ACL_FINDENTRY,  ! Locate a specific ACE
192 0191 1     ACL_FINDTYPE,   ! Locate a specific ACE type
193 0192 1     ACL_DELETEACL,  ! Delete the entire ACL
194 0193 1     ACL_READACL,     ! Read the ACL
195 0194 1     ACL_ACLLENGTH, ! Get the ACL's length
```

```
196 0195 1 ACL_READACE      Read a single ACE
197 0196 1 ACL_LOCATEACE    Locate ACE by context
198 0197 1 ACL_INIT_QUEUE Initialize the ACL queue
199 0198 1 ALLOC_PAGED     Paged pool allocator
200 0199 1 DALLOC_PAGED     Paged pool deallocator
201 0200 1 LIB$TPARSE      General purpose parser
202 0201 1 LIB$FID_TO_NAME   FID to file-spec translator
203 0202 1 LIB$GET_VM       General memory allocator
204 0203 1 LIB$FREE_VM      General memory deallocator
205 0204 1 EXESPROBER      : L_PROBE ADDRESSING_MODE (GENERAL)
206 0205 1               Probe buffer for read
207 0206 1 EXESPROBEW      : L_PROBE ADDRESSING_MODE (GENERAL)
208 0207 1               Probe buffer for write
209 0208 1 IOCSVERIFYCHAN   : L_VERIFY ADDRESSING_MODE (GENERAL)
210 0209 1               Verify channel number
211 0210 1 SCH$LOCKR     : L_Mutex ADDRESSING_MODE (GENERAL)
212 0211 1               Lock mutex for read
213 0212 1 SCH$LOCKW     : L_Mutex ADDRESSING_MODE (GENERAL)
214 0213 1               Lock mutex for write
215 0214 1 SCH$UNLOCK     : L_Mutex ADDRESSING_MODE (GENERAL)
216 0215 1               Unlock mutex
217 0216 1
218 0217 1 EXTERNAL
219 0218 1 CTL$GL_PCB       : REF $BBLOCK;      ! Address of process PCB
220 0219 1
221 0220 1 MACRO
222 0221 1   ARG_COUNT =
223 0222 1   BEGIN
224 0223 1   BUILTIN AP;
225 0224 1   (.AP)<0,8>
226 0225 1   END
227 0226 1   %
228 0227 1
229 0228 1   SET_IPL (LEVEL) =
230 0229 1   BEGIN
231 0230 1   BUILTIN MTPR;
232 0231 1   MTPR (%REF (LEVEL), PR$IPL)
233 0232 1   END
234 0233 1   %
235 0234 1
236 0235 1 LITERAL
237 0236 1   ACL_TYPE      = 7;      ! Must parallel [F11X.SRC]FCPDEF.B32
238 0237 1   MAX_ACL_SIZE = 512;   ! Max size of an ACL segment
239 0238 1
240 0239 1 LITERAL
241 0240 1   MIN_OBJECT_TYPE = MINU (ACL$C_FILE,
242 0241 1   ACL$C_DEVICE,
243 0242 1   ACL$C_JOBCTL_QUEUE,
244 0243 1   ACL$C_COMMON_EF_CLUSTER,
245 0244 1   ACL$C_LOGICAL_NAME_TABLE,
246 0245 1   ACL$C_PROCESS,
247 0246 1   ACL$C_GLOBAL_SECTION),
248 0247 1
249 0248 1   MAX_OBJECT_TYPE = MAXU (ACL$C_FILE,
250 0249 1   ACL$C_DEVICE,
251 0250 1   ACL$C_JOBCTL_QUEUE,
252 0251 1   ACL$C_COMMON_EF_CLUSTER,
```



```

: 253      0252 1      ACL$C_LOGICAL_NAME_TABLE,
: 254      0253 1      ACL$C_PROCESS,
: 255      0254 1      ACL$C_GLOBAL_SECTION),
: 256      0255 1
: 257      0256 1      MIN_ACL_ATR = MINU (ACL$C_ADDACLENT,
: 258      0257 1      ACL$C_DELACLENT,
: 259      0258 1      ACL$C_MODACLENT,
: 260      0259 1      ACL$C_FNDACLENT,
: 261      0260 1      ACL$C_FNDACETYP,
: 262      0261 1      ACL$C_DELETEACL,
: 263      0262 1      ACL$C_READACL,
: 264      0263 1      ACL$C_ACLLENGTH,
: 265      0264 1      ACL$C_READACE,
: 266      0265 1      ACL$C_RLOCK_ACL,
: 267      0266 1      ACL$C_WLOCK_ACL,
: 268      0267 1      ACL$C_UNLOCK_ACL),
: 269      0268 1
: 270      0269 1      MAX_ACL_ATR = MAXU (ACL$C_ADDACLENT,
: 271      0270 1      ACL$C_DELACLENT,
: 272      0271 1      ACL$C_MODACLENT,
: 273      0272 1      ACL$C_FNDACLENT,
: 274      0273 1      ACL$C_FNDACETYP,
: 275      0274 1      ACL$C_DELETEACL,
: 276      0275 1      ACL$C_READACL,
: 277      0276 1      ACL$C_ACLLENGTH,
: 278      0277 1      ACL$C_READACE,
: 279      0278 1      ACL$C_RLOCK_ACL,
: 280      0279 1      ACL$C_WLOCK_ACL,
: 281      0280 1      ACL$C_UNLOCK_ACL);
: 282      0281 1
: 283      0282 1      OWN
: 284      0283 1      JOURNAL_ACES : BYTE INITIAL (0),      ! Journaling ACEs allowed
: 285      0284 1      ! 0 = no support
: 286      0285 1      ! 1 = support in
: 287      0286 1      ACE_BUFFER : $BBLOCK [ATR$S_READACL],      ! Storage for binary ACE
: 288      0287 1      ACE_INDEX,      ! Index into ACE key area
: 289      0288 1      ACE_TYPE,      ! ACE type code
: 290      0289 1      ACE_RIGHTS,      ! ACE access rights
: 291      0290 1      UIC_FLAGS,      ! UIC conversion flags
: 292      0291 1      UIC_COUNT,      ! Number of UIC id's entered
: 293      0292 1      IDENTIFIER : $BBLOCK [4],      ! Converted identifier
: 294      0293 1      ID_NAME : $BBLOCK [DSC$C_S_BLN],      ! ID name descriptor
: 295      0294 1      ID_COUNT,      ! Number of identifiers given
: 296      0295 1      JOURNAL_NAME : $BBLOCK [DSC$C_S_BLN],      ! Journal name descr
: 297      0296 1      ACCESS_FLAGS,      ! Audit access flags
: 298      0297 1      SYSTEM_PROT : $BBLOCK [4],      ! System protection default
: 299      0298 1      OWNER_PROT : $BBLOCK [4],      ! Owner protection default
: 300      0299 1      GROUP_PROT : $BBLOCK [4],      ! Group protection default
: 301      0300 1      WORLD_PROT : $BBLOCK [4],      ! World protection default
: 302      0301 1      BIT_NAME_TABLE : REF BLOCKVECTOR [,DSC$C_S_BLN,BYTE],      ! Access bit name table addr
: 303      0302 1      CHANGE_ACMODE,      ! Access mode for $CHANGE_ACL
: 304      0303 1      CALL_ACMODE,      ! Access mode of caller
: 305      0304 1      PARENT_ID,      ! Parent ID for ACL locks
: 306      0305 1      ACL_QUEUE_HEAD : REF $BBLOCK,      ! Address of the ACL queue head
: 307      0306 1      ACL_POINTER : REF $BBLOCK,      ! Address of current segment
: 308      0307 1      ACL_SPLIT,      ! Offset to ACE in segment
: 309      0308 1      ACE_POINTER : REF $BBLOCK,      ! Address of current ACE
```

```

310 0309 1      ACE_NUMBER,      ! Numeric position of ACE in ACL
311 0310 1      ACL_AREA      : $BBLOCK [MAX_ACL_SIZE], ! Temp storage for ACL segment
312 0311 1      ACL_CONTEXT,    ! Context used in $CHANGE_ACL
313 0312 1      LOCK_RESNAM,    : $BBLOCK [DSC$C_S_BLN], ! Lock resource name desc
314 0313 1      RESNAM_TEXT    : $BBLOCK [31];      ! Actual resource name
315 0314 1
316 0315 1      ! Macro defining the subfields used within the resource name field.
317 0316 1
318 0317 1      MACRO
319 0318 1          RSN_T_PREFIX    = 0, 0, 0, 0 %,      ! Lock name prefix
320 0319 1          RSN_T_DEVNAM   = 8, 0, 0, 0 %,      ! Device name for device and
321 0320 1                                     ! file type objects
322 0321 1          RSN_L_FID      = 24, 0, 32, 0 %,      ! File-id for lock
323 0322 1          RSN_W_FID_NUM  = 24, 0, 16, 0 %,      ! File number
324 0323 1          RSN_W_FID_SEQ  = 26, 0, 16, 0 %,      ! File sequence number
325 0324 1
326 0325 1      LITERAL
327 0326 1          RSN_S_PREFIX    = 8,      ! Size of lock name prefix
328 0327 1          RSN_S_DEVNAM   = 16;      ! Size of device name
329 0328 1
330 0329 1      ! Assumptions made about various fields used.
331 0330 1
332 0331 1      ! The following assumptions should track the definitions in
333 0332 1      ! [RMS.SRC]RMSFILSTR.SDL module RJRDEF and
334 0333 1      ! [VMSLIB.SRC]STARDEFAE.SDL module ACEDEF
335 0334 1
336 0335 1      $ASSUME (RJR$S_JNLID EQL 28);
337 0336 1      $ASSUME ($BYTEOFFSET (RJR$T_VOLNAM) EQL 8);
338 0337 1      $ASSUME ($BYTEOFFSET (RJR$T_FID) EQL 20);
339 0338 1      $ASSUME ($BYTEOFFSET (RJR$Q_ID_DATE) EQL 28);
340 0339 1
341 0340 1      ! Define the default bit names.
342 0341 1
343 0342 1      BIND
344 0343 1          DEFAULT_BITS    = UPLIT (
345 0344 1              $DESCRIPTOR ('READ'),
346 0345 1              $DESCRIPTOR ('WRITE'),
347 0346 1              $DESCRIPTOR ('EXECUTE'),
348 0347 1              $DESCRIPTOR ('DELETE'),
349 0348 1              $DESCRIPTOR ('CONTROL'),
350 0349 1              $DESCRIPTOR ('BIT-5'),
351 0350 1              $DESCRIPTOR ('BIT-6'),
352 0351 1              $DESCRIPTOR ('BIT-7'),
353 0352 1              $DESCRIPTOR ('BIT-8'),
354 0353 1              $DESCRIPTOR ('BIT-9'),
355 0354 1              $DESCRIPTOR ('BIT-10'),
356 0355 1              $DESCRIPTOR ('BIT-11'),
357 0356 1              $DESCRIPTOR ('BIT-12'),
358 0357 1              $DESCRIPTOR ('BIT-13'),
359 0358 1              $DESCRIPTOR ('BIT-14'),
360 0359 1              $DESCRIPTOR ('BIT-15'),
361 0360 1              $DESCRIPTOR ('BIT-16'),
362 0361 1              $DESCRIPTOR ('BIT-17'),
363 0362 1              $DESCRIPTOR ('BIT-18'),
364 0363 1              $DESCRIPTOR ('BIT-19'),
365 0364 1              $DESCRIPTOR ('BIT-20'),
366 0365 1              $DESCRIPTOR ('BIT-21'),
```

```
: 367      0366 1
: 368      0367 1
: 369      0368 1
: 370      0369 1
: 371      0370 1
: 372      0371 1
: 373      0372 1
: 374      0373 1
: 375      0374 1
: 376      0375 1
: 377      0376 1
: 378      0377 1
: 379      0378 1
: 380      0379 1
: 381      0380 1
: 382      0381 1
: 383      0382 1
: 384      0383 1
: 385      0384 1
: 386      0385 1
: 387      0386 1
: 388      0387 1
```

LOCK_PREFIX

= UPLIT (

```
$DESCRIPTOR ('BIT_22'),
$DESCRIPTOR ('BIT_23'),
$DESCRIPTOR ('BIT_24'),
$DESCRIPTOR ('BIT_25'),
$DESCRIPTOR ('BIT_26'),
$DESCRIPTOR ('BIT_27'),
$DESCRIPTOR ('BIT_28'),
$DESCRIPTOR ('BIT_29'),
$DESCRIPTOR ('BIT_30'),
$DESCRIPTOR ('BIT_31')
) : VECTOR,

$DESCRIPTOR ('ACL$LOCK'),
$DESCRIPTOR ('ACL$FIL_'),
$DESCRIPTOR ('ACL$DEV_'),
$DESCRIPTOR ('ACL$JBC_'),
$DESCRIPTOR ('ACL$CEF_'),
$DESCRIPTOR ('ACL$LNT_'),
$DESCRIPTOR ('ACL$PRC_'),
$DESCRIPTOR ('ACL$GBL_')
) : VECTOR;
```


TPARSE tables for \$PARSE_ACL

```
390 0388 1 %SBTTL 'TPARSE tables for $PARSE_ACL'
391 0389 1 ! TPARSE tables to parse an Access Control List (ACL) entry.
392 0390 1
393 0391 1 $INIT_STATE (ACE_STATE, ACE_KEY);
394 0392 1
395 0393 1 ! Determine the type of ACE
396 0394 1
397 P 0395 1 $STATE (
398 P 0396 1 ('(')
399 0397 1 );
400 0398 1
401 P 0399 1 $STATE (GET_KEYWORD,
402 P 0400 1 ('IDENTIFIER', GET_ID, ACESC_KEYID, ACE_TYPE),
403 P 0401 1 ('BI_JOURNAL_NAME', GET_JNL, ACESC_BIJNL, ACE_TYPE),
404 P 0402 1 ('AI_JOURNAL_NAME', GET_JNL, ACESC_AIJNL, ACE_TYPE),
405 P 0403 1 ('AT_JOURNAL_NAME', GET_JNL, ACESC_ATJNL, ACE_TYPE),
406 P 0404 1 ('AUDIT_JOURNAL', GET_ADDIT, ACESC_AUDIT, ACE_TYPE),
407 P 0405 1 ('ALARM_JOURNAL', GET_ALARM, ACESC_ALARM, ACE_TYPE),
408 P 0406 1 ('ACCESS', GET_ACCESS),
409 P 0407 1 ('OPTIONS', GET_FLAGS),
410 P 0408 1 ('DEFAULT_PROTECTION', GET_PROT, ACESC_DIRDEF, ACE_TYPE)
411 0409 1 );
412 P 0410 1 $STATE (
413 P 0411 1 ('', GET_KEYWORD),
414 P 0412 1 (')', CHK_FOR_END)
415 0413 1 );
416 0414 1
417 0415 1 ! Access Control Entry.
418 0416 1
419 P 0417 1 $STATE (GET_ID,
420 P 0418 1 ('='),
421 P 0419 1 (':'))
422 0420 1 );
423 P 0421 1 $STATE (GET_IDTYPE,
424 P 0422 1 (TPAS_IDENT, ..., IDENTIFIER)
425 0423 1 );
426 0424 1
427 0425 1 ! Check for the end of the identifier.
428 0426 1
429 P 0427 1 $STATE (CHK_ENDID,
430 P 0428 1 ('', GET_KEYWORD, SET_ID),
431 P 0429 1 ('+', GET_IDTYPE, SET_ID),
432 P 0430 1 (')', CHK_FOR_END, SET_ID)
433 0431 1 );
434 0432 1
435 0433 1 ! RMS Journal name
436 0434 1
437 P 0435 1 $STATE (GET_JNL,
438 P 0436 1 ('='),
439 P 0437 1 (':'))
440 0438 1 );
441 P 0439 1 $STATE (
442 P 0440 1 ((GET_STRING), ..., JOURNAL_NAME)
443 0441 1 );
444 0442 1
445 0443 1 ! Check for the end of the journal name.
446 0444 1
```

TPARSE tables for \$PARSE_ACL

```
.. 447 P 0445 1 $STATE (
448 P 0446 1      ('),CHK_FOR_END)
449 P 0447 1      );
450 P 0448 1
451 P 0449 1 ! File access audit.
452 P 0450 1
453 P 0451 1 $STATE (GET_AUDIT,
454 P 0452 1      ('='),
455 P 0453 1      (':'),
456 P 0454 1      );
457 P 0455 1 $STATE (
458 P 0456 1      ((GET_STRING),...,JOURNAL_NAME)
459 P 0457 1      );
460 P 0458 1
461 P 0459 1 ! Check to see if there is an access type to follow
462 P 0460 1
463 P 0461 1 $STATE (
464 P 0462 1      ('',GET_KEYWORD),
465 P 0463 1      (')',CHK_FOR_END)
466 P 0464 1      );
467 P 0465 1
468 P 0466 1 ! File access alarm
469 P 0467 1
470 P 0468 1 $STATE (GET_ALARM,
471 P 0469 1      ('='),
472 P 0470 1      (':'),
473 P 0471 1      );
474 P 0472 1 $STATE (
475 P 0473 1      ((GET_STRING),...,JOURNAL_NAME)
476 P 0474 1      );
477 P 0475 1
478 P 0476 1 ! Check to see if there is an access type to follow
479 P 0477 1
480 P 0478 1 $STATE (
481 P 0479 1      ('',GET_KEYWORD),
482 P 0480 1      (')',CHK_FOR_END)
483 P 0481 1      );
484 P 0482 1
485 P 0483 1 ! Get the access type code
486 P 0484 1
487 P 0485 1 $STATE (GET_ACCESS,
488 P 0486 1      ('='),
489 P 0487 1      (':'),
490 P 0488 1      );
491 P 0489 1 $STATE (GET_ACCTYPE,
492 P 0490 1      ('SUCCESS',...,ACESM_SUCCESS,ACCESS_FLAGS),
493 P 0491 1      ('FAILURE',...,ACESM_FAILURE,ACCESS_FLAGS),
494 P 0492 1      ('NONE'),
495 P 0493 1      ((GET_STRING),,SET_ACCESS_BIT)
496 P 0494 1      );
497 P 0495 1 $STATE (
498 P 0496 1      ('+',GET_ACCTYPE),
499 P 0497 1      (')',CHK_FOR_END),
500 P 0498 1      ('',GET_KEYWORD)
501 P 0499 1      );
502 P 0500 1
503 P 0501 1 ! Get any special flags applied to the ACE.
```

```
504      0502 1
505      P 0503 1 $STATE (GET_FLAGS,
506      P 0504 1      ('='),
507      P 0505 1      (':'),
508      P 0506 1      );
509      P 0507 1 $STATE (GET_FLAGTYPE,
510      P 0508 1      ('DEFAULT',,,ACESM_DEFAULT,ACE_BUFFER[ACESW_FLAGS]),
511      P 0509 1      ('HIDDEN',,,ACESM_HIDDEN,ACE_BUFFER[ACESW_FLAGS]),
512      P 0510 1      ('PROTECTED',,,ACESM_PROTECTED,ACE_BUFFER[ACESW_FLAGS]),
513      P 0511 1      ('NOPROPAGATE',,,ACESM_NOPROPAGATE,ACE_BUFFER[ACESW_FLAGS]),
514      P 0512 1      ('NONE'),
515      P 0513 1      );
516      P 0514 1 $STATE (
517      P 0515 1      ('+',GET_FLAGTYPE),
518      P 0516 1      (')',CHK_FOR_END),
519      P 0517 1      (',',GET_KEYWORD),
520      P 0518 1      );
521      P 0519 1
522      P 0520 1 ! Get the directory default protection.
523      P 0521 1
524      P 0522 1 $STATE (GET_PROT,
525      P 0523 1      ('='),
526      P 0524 1      );
527      P 0525 1 $STATE (GET_PROT_CLASS,
528      P 0526 1      ('SYSTEM',GET_SYS_PRO),
529      P 0527 1      ('OWNER',GET_OWN_PRO),
530      P 0528 1      ('GROUP',GET_GRP_PRO),
531      P 0529 1      ('WORLD',GET_WOR_PRO),
532      P 0530 1      (TPAS_LAMBDA,GET_KEYWORD),
533      P 0531 1      );
534      P 0532 1 $STATE (GET_SYS_PRO,
535      P 0533 1      ('='),
536      P 0534 1      );
537      P 0535 1      (TPAS_LAMBDA,CHK_END_PRO)
538      P 0536 1      );
539      P 0537 1 $STATE (GET_SYS_PRO1,
540      P 0538 1      ('R',GET_SYS_PRO1,,ARMSM_READ,SYSTEM_PROT),
541      P 0539 1      ('W',GET_SYS_PRO1,,ARMSM_WRITE,SYSTEM_PROT),
542      P 0540 1      ('E',GET_SYS_PRO1,,ARMSM_EXECUTE,SYSTEM_PROT),
543      P 0541 1      ('D',GET_SYS_PRO1,,ARMSM_DELETE,SYSTEM_PROT),
544      P 0542 1      ('C',GET_SYS_PRO1,,ARMSM_CONTROL,SYSTEM_PROT),
545      P 0543 1      (TPAS_LAMBDA,CHK_END_PRO),
546      P 0544 1      );
547      P 0545 1 $STATE (GET_OWN_PRO,
548      P 0546 1      ('='),
549      P 0547 1      );
550      P 0548 1      (TPAS_LAMBDA,CHK_END_PRO)
551      P 0549 1      );
552      P 0550 1 $STATE (GET_OWN_PRO1,
553      P 0551 1      ('R',GET_OWN_PRO1,,ARMSM_READ,OWNER_PROT),
554      P 0552 1      ('W',GET_OWN_PRO1,,ARMSM_WRITE,OWNER_PROT),
555      P 0553 1      ('E',GET_OWN_PRO1,,ARMSM_EXECUTE,OWNER_PROT),
556      P 0554 1      ('D',GET_OWN_PRO1,,ARMSM_DELETE,OWNER_PROT),
557      P 0555 1      ('C',GET_OWN_PRO1,,ARMSM_CONTROL,OWNER_PROT),
558      P 0556 1      (TPAS_LAMBDA,CHK_END_PRO),
559      P 0557 1      );
560      P 0558 1 $STATE (GET_GRP_PRO,
```



```
561 P 0559 1 (';'),
562 P 0560 1 ('='),
563 P 0561 1 (TPAS_LAMBDA,CHK_END_PRO)
564 P 0562 1 );
565 P 0563 1 $STATE (GET_GRP_PRO1,
566 P 0564 1 ('R',GET_GRP_PRO1,,ARMSM_READ,GROUP_PROT),
567 P 0565 1 ('W',GET_GRP_PRO1,,ARMSM_WRITE,GROUP_PROT),
568 P 0566 1 ('E',GET_GRP_PRO1,,ARMSM_EXECUTE,GROUP_PROT),
569 P 0567 1 ('D',GET_GRP_PRO1,,ARMSM_DELETE,GROUP_PROT),
570 P 0568 1 ('C',GET_GRP_PRO1,,ARMSM_CONTROL,GROUP_PROT),
571 P 0569 1 (TPAS_LAMBDA,CHK_END_PRO)
572 P 0570 1 );
573 P 0571 1 $STATE (GET_WOR_PRO,
574 P 0572 1 (';'),
575 P 0573 1 ('='),
576 P 0574 1 (TPAS_LAMBDA,CHK_END_PRO)
577 P 0575 1 );
578 P 0576 1 $STATE (GET_WOR_PRO1,
579 P 0577 1 ('R',GET_WOR_PRO1,,ARMSM_READ,WORLD_PROT),
580 P 0578 1 ('W',GET_WOR_PRO1,,ARMSM_WRITE,WORLD_PROT),
581 P 0579 1 ('E',GET_WOR_PRO1,,ARMSM_EXECUTE,WORLD_PROT),
582 P 0580 1 ('D',GET_WOR_PRO1,,ARMSM_DELETE,WORLD_PROT),
583 P 0581 1 ('C',GET_WOR_PRO1,,ARMSM_CONTROL,WORLD_PROT),
584 P 0582 1 (TPAS_LAMBDA,CHK_END_PRO)
585 P 0583 1 );
586 P 0584 1 );
587 P 0585 1 $STATE (CHK_END_PRO,
588 P 0586 1 (';',GET_PROT_CLASS),
589 P 0587 1 (';'),CHK_FOR_END)
590 P 0588 1 );
591 P 0589 1 );
592 P 0590 1 ! Parse off a random string.
593 P 0591 1 );
594 P 0592 1 $STATE (GET_STRING,
595 P 0593 1 (';',TPAS_FAIL),
596 P 0594 1 (';'),TPAS_FAIL),
597 P 0595 1 (TPAS_EOS,TPAS_FAIL),
598 P 0596 1 ((GET_STRING1))
599 P 0597 1 );
600 P 0598 1 $STATE (GET_STRING1,
601 P 0599 1 ((CHK_DELIM),GET_STRING1),
602 P 0600 1 (TPAS_LAMBDA,TPAS_EXIT)
603 P 0601 1 );
604 P 0602 1 $STATE (CHK_DELIM,
605 P 0603 1 ('+',TPAS_FAIL),
606 P 0604 1 (';',TPAS_FAIL),
607 P 0605 1 (';'),TPAS_FAIL),
608 P 0606 1 (TPAS_EOS,TPAS_FAIL),
609 P 0607 1 (TPAS_ANY,TPAS_EXIT)
610 P 0608 1 );
611 P 0609 1 );
612 P 0610 1 ! Check for the end of the ACE. Trailing blanks are allowed.
613 P 0611 1 );
614 P 0612 1 $STATE (CHK_FOR_END,
615 P 0613 1 (TPAS_EOS,TPAS_EXIT),
616 P 0614 1 );
```

```
618 0615 1 XSBTTL 'SPARSE_ACL system service'
619 0616 1 GLOBAL ROUTINE SYSSPARSE_ACL (ACL_STRING, ACL_ENTRY, ERROR_POSITION, BIT_TABLE) =
620 0617 1
621 0618 1 ++
622 0619 1
623 0620 1 FUNCTIONAL DESCRIPTION:
624 0621 1
625 0622 1 This routine converts the Access Control Entry from a text form to
626 0623 1 the binary form.
627 0624 1
628 0625 1 CALLING SEQUENCE:
629 0626 1 SYSSPARSE_ACL (ARG1, ARG2, ARG3, ARG4)
630 0627 1
631 0628 1 INPUT PARAMETERS:
632 0629 1 ARG1: address of the input text descriptor
633 0630 1 ARG2: address of the output buffer descriptor
634 0631 1 ARG4: address of an access bit name table
635 0632 1
636 0633 1 IMPLICIT INPUTS:
637 0634 1 none
638 0635 1
639 0636 1 OUTPUT PARAMETERS:
640 0637 1 ARG3: number of characters processed
641 0638 1
642 0639 1 IMPLICIT OUTPUTS:
643 0640 1 none
644 0641 1
645 0642 1 ROUTINE VALUE:
646 0643 1 SSS_NORMAL: The conversion was successful.
647 0644 1 SSS_ACCVIO: The input or output descriptors cannot be read, the
648 0645 1 output buffer cannot be written, or the output buffer
649 0646 1 is not large enough to contain the converted ACE.
650 0647 1 SSS_IVACL: The syntax of the input ACE is invalid.
651 0648 1 SSS_NOSUCHID: The identifier specified in the ACE is not in the
652 0649 1 rights database.
653 0650 1
654 0651 1 SIDE EFFECTS:
655 0652 1 none
656 0653 1
657 0654 1 --
658 0655 1
659 0656 2 BEGIN
660 0657 2
661 0658 2 MAP
662 0659 2 ACL_STRING : REF $BBLOCK, ! Address of input descriptor
663 0660 2 ACL_ENTRY : REF $BBLOCK, ! Address of output descriptor
664 0661 2 ERROR_POSITION : REF VECTOR [,WORD], ! Syntax error position
665 0662 2 BIT_TABLE : REF VECTOR; ! Address of access bit name table
666 0663 2
667 0664 2 LOCAL
668 0665 2 STATUS, ! Routine exit status
669 0666 2 ACL_STRING_LEN, ! Length of input ACL string
670 0667 2 ACL_ENTRY_LEN, ! Length of output ACE buffer
671 0668 2 TPARSE_BLOCK : $BBLOCK [TPASK_LENGTH0]; ! Parser context block
672 0669 2
673 0670 2 EXTERNAL LITERAL
674 0671 2 LIB$SYNTAXERR;
```

```
675 0672 2
676 0673 2 ! Check to see if an access bit name table was supplied. If so, use it
677 0674 2 ! rather than the default table.
678 0675 2
679 0676 2 BIT_NAME_TABLE = 0;
680 0677 2 IF .BIT_TABLE NEQA 0
681 0678 2 THEN IF PROBER (%REF (0), %REF (256), .BIT_TABLE)
682 0679 2 THEN BIT_NAME_TABLE = .BIT_TABLE
683 0680 2 ELSE RETURN SSS_ACCVIO;
684 0681 2
685 0682 2 ! Set up initial parameters.
686 0683 2
687 0684 2 CH$FILL (0, ATR$S_READACL, ACE_BUFFER);
688 0685 2 ACE_INDEX = ACE_TYPE = ACE_RIGHTS = 0;
689 0686 2 UIC_FLAGS = UIC_COUNT = 0;
690 0687 2 ID_COUNT = 0;
691 0688 2 ACCESS_FLAGS = 0;
692 0689 2 JOURNAL_NAME[DSC$W_LENGTH] = ID_NAME[DSC$W_LENGTH] = 0;
693 0690 2 SYSTEM_PROT = OWNER_PROT = GROUP_PROT = WORLD_PROT = 0;
694 0691 2
695 0692 2 CH$FILL (0, TPASK_LENGTH0, TPARSE_BLOCK);
696 0693 2 TPARSE_BLOCK[TPAS$ COUNT] = TPASK_COUNT0;
697 0694 2 TPARSE_BLOCK[TPAS$ ABBREV] = 1;
698 0695 2 IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .ACL_STRING)
699 0696 2 THEN
700 0697 2 BEGIN
701 0698 2 ACL_STRING_LEN = TPARSE_BLOCK[TPAS$ STRINGCNT] = .ACL_STRING[DSC$W_LENGTH];
702 0699 2 IF EX$PROBER (0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
703 0700 2 THEN TPARSE_BLOCK[TPAS$ STRINGPTR] = .ACL_STRING[DSC$A_POINTER]
704 0701 2 ELSE RETURN SSS_ACCVIO;
705 0702 2 END
706 0703 2 ELSE RETURN SSS_ACCVIO;
707 0704 2
708 0705 2 STATUS = LIB$TPARSE (TPARSE_BLOCK, ACE_STATE, ACE_KEY);
709 0706 2
710 0707 2 ! If necessary set the number of characters processed.
711 0708 2
712 0709 2 IF .ERROR_POSITION NEQA 0
713 0710 2 THEN IF PROBER (%REF (0), %REF (2), .ERROR_POSITION)
714 0711 2 THEN ERROR_POSITION[0] = .ACL_STRING[EN -
715 0712 2 .TPARSE_BLOCK[TPAS$ STRINGCNT]
716 0713 2 ELSE RETURN SSS_ACCVIO;
717 0714 2
718 0715 2 ! If there 1) are any syntax errors, 2) is an invalid ACE type (zero),
719 0716 2 ! or 3) is remaining text to the ACE; return an error.
720 0717 2
721 0718 2 IF .STATUS EQL LIB$_SYNTAXERR
722 0719 2 OR .ACE_TYPE EQL 0
723 0720 2 OR (.STATUS AND .TPARSE_BLOCK[TPAS$ STRINGCNT] GTR 0)
724 0721 2 THEN RETURN SSS_IVACL;
725 0722 2
726 0723 2 IF NOT .STATUS THEN RETURN .STATUS;
727 0724 2
728 0725 2 ! Set up the standard ACE fields.
729 0726 2
730 0727 2 ACE_BUFFER[ACES$B_TYPE] = .ACE_TYPE;
731 0728 2 ACE_BUFFER[ACES$W_FLAGS] = .ACE_BUFFER[ACES$W_FLAGS]
```



```
0732      OR .UIC_FLAGS  
0733      OR .ACCESS_FLAGS;  
0734      ACE_BUFFER[ACESL_ACCESS] = .ACE_RIGHTS;  
0735  
0736      ! Based upon the type code, finish up the ACE. Then do the final error  
0737      ! checking to make sure that I didn't get more than I wanted.  
0738  
0739      CASE .ACE_TYPE FROM ACESC_KEYID TO ACESC_DIRDEF OF  
0740      SET  
0741      [ACESC_KEYID]:  
0742      BEGIN  
0743      IF .ACCESS_FLAGS NEQ 0  
0744      OR .JOURNAL_NAME[DSCSW_LENGTH] NEQ 0  
0745      OR .UIC_COUNT GTR 1  
0746      OR .ACE_INDEX EQL 0  
0747      THEN RETURN SSS_IVACL;  
0748      ACE_BUFFER[ACESB_SIZE] = ACESC_LENGTH + .ACE_INDEX * 4;  
0749      END;  
0750      [ACESC_BIJNL,  
0751      ACESC-AIJNL,  
0752      ACESC-ATJNL]:  
0753      BEGIN  
0754      IF NOT .JOURNAL ACES THEN RETURN SSS_IVACL;  
0755      IF .UIC_COUNT NEQ 0  
0756      OR .ID COUNT NEQ 0  
0757      OR .ACCESS_FLAGS NEQ 0  
0758      OR .ACE_RIGHTS NEQ 0  
0759      THEN RETURN SSS_IVACL;  
0760      CHSMOVE (.JOURNAL_NAME[DSCSW_LENGTH],  
0761      .JOURNAL_NAME[DSCSA_POINTER],  
0762      ACE_BUFFER[ACESL_ACCESS]);  
0763      ACE_BUFFER[ACESB_SIZE] = .JOURNAL_NAME[DSCSW_LENGTH] +  
0764      $BYTEOFFSET (ACESL_ACCESS);  
0765      END;  
0766      [ACESC_AUDIT,  
0767      ACESC-ALARM]:  
0768      BEGIN  
0769      IF .UIC_COUNT NEQ 0  
0770      OR .ID COUNT NEQ 0  
0771      OR .JOURNAL_NAME[DSCSW_LENGTH] NEQ %CHARCOUNT ('SECURITY')  
0772      OR CH$NEQ (%CHARCOUNT ('SECURITY'), UPLIT ('SECURITY'),  
0773      .JOURNAL_NAME[DSCSW_LENGTH], .JOURNAL_NAME[DSCSA_POINTER], 0)  
0774      THEN RETURN SSS_IVACL;  
0775      CHSMOVE (.JOURNAL_NAME[DSCSW_LENGTH],  
0776      .JOURNAL_NAME[DSCSA_POINTER],  
0777      ACE_BUFFER[ACESL_KEY]);  
0778      ACE_BUFFER[ACESB_SIZE] = ACESC_LENGTH + .JOURNAL_NAME[DSCSW_LENGTH];  
0779      END;  
0780      [ACESC_DIRDEF]:  
0781      BEGIN  
0782      IF .ACCESS_FLAGS NEQ 0  
0783      OR .JOURNAL_NAME[DSCSW_LENGTH] NEQ 0  
0784      OR .UIC_COUNT NEQ 0  
0785      OR .ID COUNT NEQ 0  
0786      THEN RETURN SSS_IVACL;  
0787      SYSTEM_PROT = NOT .SYSTEM_PROT;  
0788      SYSTEM_PROT[ARMSV_FILL] = 0;
```

```

789 0786      ACE_BUFFER[ACESL_SYS_PROT] = .SYSTEM_PROT;
790 0787      OWNER_PROT = NOT .OWNER_PROT;
791 0788      OWNER_PROT[ARMSV_FILL] = 0;
792 0789      ACE_BUFFER[ACESL_OWN_PROT] = .OWNER_PROT;
793 0790      GROUP_PROT = NOT .GROUP_PROT;
794 0791      GROUP_PROT[ARMSV_FILL] = 0;
795 0792      ACE_BUFFER[ACESL_GRP_PROT] = .GROUP_PROT;
796 0793      WORLD_PROT = NOT .WORLD_PROT;
797 0794      WORLD_PROT[ARMSV_FILL] = 0;
798 0795      ACE_BUFFER[ACESL_WOR_PROT] = .WORLD_PROT;
799 0796      ACE_BUFFER[ACESB_SIZE] = ACESC_LENGTH + 16;
800 0797      END;
801 0798      [INRANGE,
802 0799      OUTRANGE]: RETURN SS$_IVACL;
803 0800      TES;
804 0801
805 0802      ! Check to make sure there is room to receive the ACE.
806 0803
807 0804      IF PROBER (%REF (0), %REF (DSC$_S_BLN), .ACL_ENTRY)
808 0805      THEN
809 0806          BEGIN
810 0807              ACL_ENTRY_LEN = .ACL_ENTRY[DSC$_LENGTH];
811 0808              IF .ACE_BUFFER[ACESB_SIZE] LEQU .ACL_ENTRY_LEN
812 0809              AND EXE$PROBEW (0, .ACL_ENTRY_LEN, .ACL_ENTRY[DSC$_POINTER])
813 0810              THEN CH$COPY (.ACE_BUFFER[ACESB_SIZE], ACE_BUFFER, 0,
814 0811                          .ACL_ENTRY[DSC$_LENGTH], .ACL_ENTRY[DSC$_POINTER])
815 0812              ELSE RETURN SS$_ACCVIO;
816 0813              END
817 0814      ELSE RETURN SS$_ACCVIO;
818 0815
819 0816      RETURN SS$_NORMAL;
820 0817
821 0818      1 END;

```

! End of routine SYS\$PARSE_ACL

.TITLE SYSACLSRV
.IDENT \V04-000\
.PSECT _LIB\$KEY1\$,NOWRT, SHR, PIC,1

```

52 45 49 46 49 54 4E 45 44 49 00000 :TPASKEYSTO
                                U.3: .BLKB 0
                                :TPASKEYST
                                U.5: .ASCII \IDENTIFIER\
                                FF 0000A .BYTE -1
                                0000B :TPASKEYSTO
                                U.11: .BLKB 0
45 4D 41 4E 5F 4C 41 4E 52 55 4F 4A 5F 49 42 0000B :TPASKEYST
                                U.13: .ASCII \BI_JOURNAL_NAME\
                                FF 0001A .BYTE -1
                                0001B :TPASKEYSTO
                                U.19: .BLKB 0
45 4D 41 4E 5F 4C 41 4E 52 55 4F 4A 5F 49 41 0001B :TPASKEYST
                                U.21: .ASCII \AI_JOURNAL_NAME\
                                FF 0002A .BYTE -1
                                0002B :TPASKEYSTO
                                U.26: .BLKB 0

```

| | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|---------------|----------------------|---|---|
| 45 | 4D | 41 | 4E | 5F | 4C | 41 | 4E | 52 | 55 | 4F | 4A | 5F | 54 | 41 | 0002B | :TPASKEYST | | | |
| | | | | | | | | | | | | | | FF | 0003A | U.28: .ASCII | \AT_JOURNAL_NAME\ | : | |
| | | | | | | | | | | | | | | | 0003B | :TPASKEYSTO | -1 | : | |
| | | | | | | | | | | | | | | | | U.33: .BLKB | 0 | : | |
| | | 4C | 41 | 4E | 52 | 55 | 4F | 4A | 5F | 54 | 49 | 44 | 55 | 41 | 0003B | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | FF | 00048 | U.35: .ASCII | \AUDIT_JOURNAL\ | : | |
| | | | | | | | | | | | | | | | 00049 | :TPASKEYSTO | -1 | : | |
| | | | | | | | | | | | | | | | | U.41: .BLKB | 0 | : | |
| | | 4C | 41 | 4E | 52 | 55 | 4F | 4A | 5F | 4D | 52 | 41 | 4C | 41 | 00049 | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | FF | 00056 | U.43: .ASCII | \ALARM_JOURNAL\ | : | |
| | | | | | | | | | | | | | | | 00057 | :TPASKEYSTO | -1 | : | |
| | | | | | | | | | | | | | | | | U.49: .BLKB | 0 | : | |
| | | | | | | 53 | 53 | 45 | 43 | 43 | | | | 41 | 00057 | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | FF | 0005D | U.51: .ASCII | \ACCESS\ | : | |
| | | | | | | | | | | | | | | | 0005E | :TPASKEYSTO | -1 | : | |
| | | | | | | | | | | | | | | | | U.55: .BLKB | 0 | : | |
| | | | | | | 53 | 4E | 4F | 49 | 54 | 50 | | | 4F | 0005E | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | FF | 00065 | U.57: .ASCII | \OPTIONS\ | : | |
| | | | | | | | | | | | | | | | 00066 | :TPASKEYSTO | -1 | : | |
| | | | | | | | | | | | | | | | | U.61: .BLKB | 0 | : | |
| 54 | 43 | 45 | 54 | 4F | 52 | 50 | 5F | 54 | 4C | 55 | 41 | 46 | 45 | 44 | 00066 | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | | | U.63: .ASCII | \DEFAULT_PROTECTION\ | : | |
| | | | | | | | | | | | | 4E | 4F | 49 | 00075 | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | FF | 00078 | :TPASKEYFILL | -1 | : | |
| | | | | | | | | | | | | | | FF | 00079 | U.69: .BYTE | -1 | : | |
| | | | | | | | | | | | | | | | 0007A | :TPASKEYSTO | | : | |
| | | | | | | | | | | | | | | | | U.116: .BLKB | 0 | : | |
| | | | | | | 53 | 53 | 45 | 43 | 43 | 55 | | | 53 | 0007A | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | FF | 00081 | U.118: .ASCII | \SUCCESS\ | : | |
| | | | | | | | | | | | | | | | 00082 | :TPASKEYSTO | -1 | : | |
| | | | | | | | | | | | | | | | | U.122: .BLKB | 0 | : | |
| | | | | | | 45 | 52 | 55 | 4C | 49 | 41 | | | 46 | 00082 | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | FF | 00089 | U.124: .ASCII | \FAILURE\ | : | |
| | | | | | | | | | | | | | | | 0008A | :TPASKEYSTO | -1 | : | |
| | | | | | | | | | | | | | | | | U.128: .BLKB | 0 | : | |
| | | | | | | | | | | | | 45 | 4E | 4F | 4E | 0008A | :TPASKEYST | | : |
| | | | | | | | | | | | | | | | | U.130: .ASCII | \NONE\ | : | |
| | | | | | | | | | | | | | | FF | 0008E | :TPASKEYFILL | -1 | : | |
| | | | | | | | | | | | | | | FF | 0008F | U.135: .BYTE | -1 | : | |
| | | | | | | | | | | | | | | | 00090 | :TPASKEYSTO | | : | |
| | | | | | | | | | | | | | | | | U.144: .BLKB | 0 | : | |
| | | | | | | 54 | 4C | 55 | 41 | 46 | 45 | | | 44 | 00090 | :TPASKEYST | | : | |
| | | | | | | | | | | | | | | FF | 00097 | U.146: .ASCII | \DEFAULT\ | : | |
| | | | | | | | | | | | | | | | 00098 | :TPASKEYSTO | -1 | : | |
| | | | | | | | | | | | | | | | | U.150: .BLKB | 0 | : | |
| | | | | | | | | | | | | | | | | U.152: .ASCII | \HIDDEN\ | : | |


```

FF 0009E :TPASKEYSTO .BYTE -1
0009F U.156: .BLKB 0
44 45 54 43 45 54 4F 52 50 0009F :TPASKEYST
U.158: .ASCII \PROTECTED\
FF 000A8 :TPASKEYSTO .BYTE -1
000A9 U.162: .BLKB 0
45 54 41 47 41 50 4F 52 50 4F 4E 000A9 :TPASKEYST
U.164: .ASCII \NOPROPAGATE\
FF 000B4 :TPASKEYSTO .BYTE -1
000B5 U.168: .BLKB 0
45 4E 4F 4E 000B5 :TPASKEYST
U.170: .ASCII \NONE\
FF 000B9 :TPASKEYFILL .BYTE -1
FF 000BA U.172: .BYTE -1
000BB :TPASKEYSTO .BLKB 0
4D 45 54 53 59 53 000BB :TPASKEYST
U.182: .ASCII \SYSTEM\
FF 000C1 :TPASKEYSTO .BYTE -1
000C2 U.186: .BLKB 0
52 45 4E 57 4F 000C2 :TPASKEYST
U.188: .ASCII \OWNER\
FF 000C7 :TPASKEYSTO .BYTE -1
000C8 U.192: .BLKB 0
50 55 4F 52 47 000C8 :TPASKEYST
U.194: .ASCII \GROUP\
FF 000CD :TPASKEYSTO .BYTE -1
000CE U.198: .BLKB 0
44 4C 52 4F 57 000CE :TPASKEYST
U.200: .ASCII \WORLD\
FF 000D3 :TPASKEYFILL .BYTE -1
FF 000D4 U.206: .BYTE -1

.PSECT _LIB$STATES, NOWRT, SHR, PIC, 1

00000 ACE_STATE:: .BLKB 0
0428 00000 :TPASTYPE U.2: .WORD 1064
00002 GET_KEYWORD: .BLKB 0
7100 00002 :TPASTYPE U.6: .WORD 28928
00000000* 00004 :TPASADDR U.7: .LONG <<ACE_TYPE-U.7>-4>
00000001 00008 :TPASMASK U.8: .LONG 1
0000* 0000C :TPASTARGET U.10: .WORD <<U.9-U.10>-2>
```

| | | | |
|-----------|-------|-------------|---------------------|
| 7101 | 0000E | :TPASTYPE | |
| | | U.14: .WORD | 28929 |
| 00000000* | 00010 | :TPASADDR | |
| | | U.15: .LONG | <<ACE_TYPE-U.15>-4> |
| 00000002 | 00014 | :TPASMASK | |
| | | U.16: .LONG | 2 |
| 0000* | 00018 | :TPASTARGET | |
| | | U.18: .WORD | <<U.17-U.18>-2> |
| 7102 | 0001A | :TPASTYPE | |
| | | U.22: .WORD | 28930 |
| 00000000* | 0001C | :TPASADDR | |
| | | U.23: .LONG | <<ACE_TYPE-U.23>-4> |
| 00000003 | 00020 | :TPASMASK | |
| | | U.24: .LONG | 3 |
| 0000* | 00024 | :TPASTARGET | |
| | | U.25: .WORD | <<U.17-U.25>-2> |
| 7103 | 00026 | :TPASTYPE | |
| | | U.29: .WORD | 28931 |
| 00000000* | 00028 | :TPASADDR | |
| | | U.30: .LONG | <<ACE_TYPE-U.30>-4> |
| 00000004 | 0002C | :TPASMASK | |
| | | U.31: .LONG | 4 |
| 0000* | 00030 | :TPASTARGET | |
| | | U.32: .WORD | <<U.17-U.32>-2> |
| 7104 | 00032 | :TPASTYPE | |
| | | U.36: .WORD | 28932 |
| 00000000* | 00034 | :TPASADDR | |
| | | U.37: .LONG | <<ACE_TYPE-U.37>-4> |
| 00000005 | 00038 | :TPASMASK | |
| | | U.38: .LONG | 5 |
| 0000* | 0003C | :TPASTARGET | |
| | | U.40: .WORD | <<U.39-U.40>-2> |
| 7105 | 0003E | :TPASTYPE | |
| | | U.44: .WORD | 28933 |
| 00000000* | 00040 | :TPASADDR | |
| | | U.45: .LONG | <<ACE_TYPE-U.45>-4> |
| 00000006 | 00044 | :TPASMASK | |
| | | U.46: .LONG | 6 |
| 0000* | 00048 | :TPASTARGET | |
| | | U.48: .WORD | <<U.47-U.48>-2> |
| 1106 | 0004A | :TPASTYPE | |
| | | U.52: .WORD | 4358 |
| 0000* | 0004C | :TPASTARGET | |
| | | U.54: .WORD | <<U.53-U.54>-2> |
| 1107 | 0004E | :TPASTYPE | |
| | | U.58: .WORD | 4359 |
| 0000* | 00050 | :TPASTARGET | |
| | | U.60: .WORD | <<U.59-U.60>-2> |
| 7508 | 00052 | :TPASTYPE | |
| | | U.64: .WORD | 29960 |
| 00000000* | 00054 | :TPASADDR | |
| | | U.65: .LONG | <<ACE_TYPE-U.65>-4> |
| 00000000* | 00058 | :TPASMASK | |
| | | U.66: .LONG | 9 |
| 0000* | 0005C | :TPASTARGET | |
| | | U.68: .WORD | <<U.67-U.68>-2> |
| 102C | 0005E | :TPASTYPE | |

| | | | | |
|-----------|-------|--------------|-------------------------|---|
| 0000* | 00060 | U.70: .WORD | 4140 | : |
| | | :TPASTARGET | | : |
| 1429 | 00062 | U.71: .WORD | <<GET_KEYWORD-U.71>-2> | : |
| | | :TPASTYPE | | : |
| | | U.72: .WORD | 5161 | : |
| 0000* | 00064 | :TPASTARGET | | : |
| | | U.74: .WORD | <<U.73-U.74>-2> | : |
| | 00066 | :GET_ID | | : |
| | | U.9: .BLKB | 0 | : |
| 003D | 00066 | :TPASTYPE | | : |
| | | U.75: .WORD | 61 | : |
| 043A | 00068 | :TPASTYPE | | : |
| | | U.76: .WORD | 1082 | : |
| | 0006A | :GET_IDTYPE: | | : |
| | | .BLKB | 0 | : |
| 45EC | 0006A | :TPASTYPE | | : |
| | | U.77: .WORD | 17900 | : |
| 00000000* | 0006C | :TPASADDR | | : |
| | | U.78: .LONG | <<IDENTIFIER-U.78>-4> | : |
| | 00070 | :CHK_ENDID: | | : |
| | | .BLKB | 0 | : |
| 902C | 00070 | :TPASTYPE | | : |
| | | U.79: .WORD | -28628 | : |
| 00000000V | 00072 | :TPASACTION | | : |
| | | U.80: .LONG | <<SET_ID-U.80>-4> | : |
| | 00076 | :TPASTARGET | | : |
| | | U.81: .WORD | <<GET_KEYWORD-U.81>-2> | : |
| 902B | 00078 | :TPASTYPE | | : |
| | | U.82: .WORD | -28629 | : |
| 00000000V | 0007A | :TPASACTION | | : |
| | | U.83: .LONG | <<SET_ID-U.83>-4> | : |
| | 0007E | :TPASTARGET | | : |
| | | U.84: .WORD | <<GET_IDTYPE-U.84>-2> | : |
| 9429 | 00080 | :TPASTYPE | | : |
| | | U.85: .WORD | -27607 | : |
| 00000000V | 00082 | :TPASACTION | | : |
| | | U.86: .LONG | <<SET_ID-U.86>-4> | : |
| | 00086 | :TPASTARGET | | : |
| | | U.87: .WORD | <<U.73-U.87>-2> | : |
| | 00088 | :GET_JNL | | : |
| | | U.17: .BLKB | 0 | : |
| 003D | 00088 | :TPASTYPE | | : |
| | | U.88: .WORD | 61 | : |
| 043A | 0008A | :TPASTYPE | | : |
| | | U.89: .WORD | 1082 | : |
| 4DF8 | 0008C | :TPASTYPE | | : |
| | | U.90: .WORD | 19960 | : |
| | 0008E | :TPASUBEXP | | : |
| | | U.92: .WORD | <<U.91-U.92>-2> | : |
| 00000000* | 00090 | :TPASADDR | | : |
| | | U.93: .LONG | <<JOURNAL_NAME-U.93>-4> | : |
| 1429 | 00094 | :TPASTYPE | | : |
| | | U.94: .WORD | 5161 | : |
| | 00096 | :TPASTARGET | | : |
| | | U.95: .WORD | <<U.73-U.95>-2> | : |
| | 00098 | :GET_AUDIT | | : |
| | | U.39: .BLKB | 0 | : |

| | | | | |
|-----------|-------|---------------|------|--------------------------|
| 003D | 00098 | :TPASTYPE | | |
| | | U.96: | WORD | 61 |
| 043A | 0009A | :TPASTYPE | | |
| | | U.97: | WORD | 1082 |
| 4DF8 | 0009C | :TPASTYPE | | |
| | | U.98: | WORD | 19960 |
| 0000* | 0009E | :TPASSUBEXP | | |
| | | U.99: | WORD | <<U.91-U.99>-2> |
| 00000000* | 000A0 | :TPASADDR | | |
| | | U.100: | LONG | <<JOURNAL_NAME-U.100>-4> |
| 102C | 000A4 | :TPASTYPE | | |
| | | U.101: | WORD | 4140 |
| 0000* | 000A6 | :TPASTARGET | | |
| | | U.102: | WORD | <<GET_KEYWORD-U.102>-2> |
| 1429 | 000A8 | :TPASTYPE | | |
| | | U.103: | WORD | 5161 |
| 0000* | 000AA | :TPASTARGET | | |
| | | U.104: | WORD | <<U.73-U.104>-2> |
| | 000AC | :GET_ALARM | | |
| | | U.47: | BLKB | 0 |
| 003D | 000AC | :TPASTYPE | | |
| | | U.105: | WORD | 61 |
| 043A | 000AE | :TPASTYPE | | |
| | | U.106: | WORD | 1082 |
| 4DF8 | 000B0 | :TPASTYPE | | |
| | | U.107: | WORD | 19960 |
| 0000* | 000B2 | :TPASSUBEXP | | |
| | | U.108: | WORD | <<U.91-U.108>-2> |
| 00000000* | 000B4 | :TPASADDR | | |
| | | U.109: | LONG | <<JOURNAL_NAME-U.109>-4> |
| 102C | 000B8 | :TPASTYPE | | |
| | | U.110: | WORD | 4140 |
| 0000* | 000BA | :TPASTARGET | | |
| | | U.111: | WORD | <<GET_KEYWORD-U.111>-2> |
| 1429 | 000BC | :TPASTYPE | | |
| | | U.112: | WORD | 5161 |
| 0000* | 000BE | :TPASTARGET | | |
| | | U.113: | WORD | <<U.73-U.113>-2> |
| | 000C0 | :GET_ACCESS | | |
| | | U.53: | BLKB | 0 |
| 003D | 000C0 | :TPASTYPE | | |
| | | U.114: | WORD | 61 |
| 043A | 000C2 | :TPASTYPE | | |
| | | U.115: | WORD | 1082 |
| | 000C4 | :GET_ACCTYPE: | | |
| | | BLKB | | 0 |
| 6109 | 000C4 | :TPASTYPE | | |
| | | U.119: | WORD | 24841 |
| 00000000* | 000C6 | :TPASADDR | | |
| | | U.120: | LONG | <<ACCESS_FLAGS-U.120>-4> |
| 00000001 | 000CA | :TPASMASK | | |
| | | U.121: | LONG | 1 |
| 610A | 000CE | :TPASTYPE | | |
| | | U.125: | WORD | 24842 |
| 00000000* | 000D0 | :TPASADDR | | |
| | | U.126: | LONG | <<ACCESS_FLAGS-U.126>-4> |
| 00000002 | 000D4 | :TPASMASK | | |

| | | | | |
|-----------|-------|----------------|----------------------------|---|
| 010B | 000D8 | U.127: .LONG | 2 | : |
| | | :TPASTYPE | | : |
| 8DF8 | 000DA | U.131: .WORD | 267 | : |
| | | :TPASTYPE | | : |
| 0000* | 000DC | U.132: .WORD | -29192 | : |
| | | :TPASSUBEXP | | : |
| 00000000V | 000DE | U.133: .WORD | <<U.91-U.133>-2> | : |
| | | :TPASACTION | | : |
| 102B | 000E2 | U.134: .LONG | <<SET_ACCESS_BIT-U.134>-4> | : |
| | | :TPASTYPE | | : |
| 0000* | 000E4 | U.136: .WORD | 4139 | : |
| | | :TPASTARGET | | : |
| 1029 | 000E6 | U.137: .WORD | <<GET_ACCTYPE-U.137>-2> | : |
| | | :TPASTYPE | | : |
| 0000* | 000E8 | U.138: .WORD | 4137 | : |
| | | :TPASTARGET | | : |
| 142C | 000EA | U.139: .WORD | <<U.73-U.139>-2> | : |
| | | :TPASTYPE | | : |
| 0000* | 000EC | U.140: .WORD | 5164 | : |
| | | :TPASTARGET | | : |
| | 000EE | U.141: .WORD | <<GET_KEYWORD-U.141>-2> | : |
| | | :GET_FLAGS | | : |
| 003D | 000EE | U.59: .BLKB | 0 | : |
| | | :TPASTYPE | | : |
| 043A | 000F0 | U.142: .WORD | 61 | : |
| | | :TPASTYPE | | : |
| | 000F2 | U.143: .WORD | 1082 | : |
| | | :GET_FLAGTYPE: | | : |
| | | .BLKB | 0 | : |
| 610C | 000F2 | U.147: .WORD | 24844 | : |
| | | :TPASTYPE | | : |
| 00000000* | 000F4 | U.148: .LONG | <<<ACE_BUFFER+2>-U.148>-4> | : |
| | | :TPASADDR | | : |
| 00000100 | 000F8 | U.149: .LONG | 256 | : |
| | | :TPASMASK | | : |
| 610D | 000FC | U.153: .WORD | 24845 | : |
| | | :TPASTYPE | | : |
| 00000000* | 000FE | U.154: .LONG | <<<ACE_BUFFER+2>-U.154>-4> | : |
| | | :TPASADDR | | : |
| 00000400 | 00102 | U.155: .LONG | 1024 | : |
| | | :TPASMASK | | : |
| 610E | 00106 | U.159: .WORD | 24846 | : |
| | | :TPASTYPE | | : |
| 00000000* | 00108 | U.160: .LONG | <<<ACE_BUFFER+2>-U.160>-4> | : |
| | | :TPASADDR | | : |
| 00000200 | 0010C | U.161: .LONG | 512 | : |
| | | :TPASMASK | | : |
| 610F | 00110 | U.165: .WORD | 24847 | : |
| | | :TPASTYPE | | : |
| 00000000* | 00112 | U.166: .LONG | <<<ACE_BUFFER+2>-U.166>-4> | : |
| | | :TPASADDR | | : |
| 00000800 | 00116 | U.167: .LONG | 2048 | : |
| | | :TPASMASK | | : |
| 0510 | 0011A | U.171: .WORD | 1296 | : |
| | | :TPASTYPE | | : |
| 102B | 0011C | U.173: .WORD | 4139 | : |

| | | | | | | |
|-----------|-------|------------------|------|--------------------------|--|---|
| 0000* | 0011E | :TPASTARGET | | | | |
| | | U.174: | WORD | <<GET_FLAGTYPE-U.174>-2> | | : |
| 1029 | 00120 | :TPASTYPE | | | | : |
| | | U.175: | WORD | 4137 | | : |
| 0000* | 00122 | :TPASTARGET | | | | : |
| | | U.176: | WORD | <<U.73-U.176>-2> | | : |
| 142C | 00124 | :TPASTYPE | | | | : |
| | | U.177: | WORD | 5164 | | : |
| 0000* | 00126 | :TPASTARGET | | | | : |
| | | U.178: | WORD | <<GET_KEYWORD-U.178>-2> | | : |
| | 00128 | :GET_PROT | | | | : |
| | | U.67: | BLKB | 0 | | : |
| 042C | 00128 | :TPASTYPE | | | | : |
| | | U.179: | WORD | 1068 | | : |
| | 0012A | :GET_PROT_CLASS: | | | | : |
| | | U.180: | BLKB | 0 | | : |
| 1111 | 0012A | :TPASTYPE | | | | : |
| | | U.183: | WORD | 4369 | | : |
| 0000* | 0012C | :TPASTARGET | | | | : |
| | | U.185: | WORD | <<U.184-U.185>-2> | | : |
| 1112 | 0012E | :TPASTYPE | | | | : |
| | | U.189: | WORD | 4370 | | : |
| 0000* | 00130 | :TPASTARGET | | | | : |
| | | U.191: | WORD | <<U.190-U.191>-2> | | : |
| 1113 | 00132 | :TPASTYPE | | | | : |
| | | U.195: | WORD | 4371 | | : |
| 0000* | 00134 | :TPASTARGET | | | | : |
| | | U.197: | WORD | <<U.196-U.197>-2> | | : |
| 1114 | 00136 | :TPASTYPE | | | | : |
| | | U.201: | WORD | 4372 | | : |
| 0000* | 00138 | :TPASTARGET | | | | : |
| | | U.203: | WORD | <<U.202-U.203>-2> | | : |
| 15F6 | 0013A | :TPASTYPE | | | | : |
| | | U.204: | WORD | 5622 | | : |
| 0000* | 0013C | :TPASTARGET | | | | : |
| | | U.205: | WORD | <<GET_KEYWORD-U.205>-2> | | : |
| | 0013E | :GET_SYS_PRO | | | | : |
| | | U.184: | BLKB | 0 | | : |
| 003A | 0013E | :TPASTYPE | | | | : |
| | | U.207: | WORD | 58 | | : |
| 003D | 00140 | :TPASTYPE | | | | : |
| | | U.208: | WORD | 61 | | : |
| 15F6 | 00142 | :TPASTYPE | | | | : |
| | | U.209: | WORD | 5622 | | : |
| 0000* | 00144 | :TPASTARGET | | | | : |
| | | U.211: | WORD | <<U.210-U.211>-2> | | : |
| | 00146 | :GET_SYS_PRO1: | | | | : |
| | | U.212: | BLKB | 0 | | : |
| 7052 | 00146 | :TPASTYPE | | | | : |
| | | U.212: | WORD | 28754 | | : |
| 00000000* | 00148 | :TPASADDR | | | | : |
| | | U.213: | LONG | <<SYSTEM_PROT-U.213>-4> | | : |
| 00000001 | 0014C | :TPASMASK | | | | : |
| | | U.214: | LONG | 1 | | : |
| 0000* | 00150 | :TPASTARGET | | | | : |
| | | U.215: | WORD | <<GET_SYS_PRO1-U.215>-2> | | : |
| 7057 | 00152 | :TPASTYPE | | | | : |

| | | | | |
|-----------|-------|----------------|--------------------------|---|
| 00000000* | 00154 | U.216: .WORD | 28759 | : |
| | | :TPASADDR | | : |
| 00000002 | 00158 | U.217: .LONG | <<SYSTEM_PROT-U.217>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 0015C | U.218: .LONG | 2 | : |
| | | :TPASTARGET | | : |
| 7045 | 0015E | U.219: .WORD | <<GET_SYS_PRO1-U.219>-2> | : |
| | | :TPASTYPE | | : |
| 00000000* | 00160 | U.220: .WORD | 28741 | : |
| | | :TPASADDR | | : |
| 00000004 | 00164 | U.221: .LONG | <<SYSTEM_PROT-U.221>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 00168 | U.222: .LONG | 4 | : |
| | | :TPASTARGET | | : |
| 7044 | 0016A | U.223: .WORD | <<GET_SYS_PRO1-U.223>-2> | : |
| | | :TPASTYPE | | : |
| 00000000* | 0016C | U.224: .WORD | 28740 | : |
| | | :TPASADDR | | : |
| 00000008 | 00170 | U.225: .LONG | <<SYSTEM_PROT-U.225>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 00174 | U.226: .LONG | 8 | : |
| | | :TPASTARGET | | : |
| 7043 | 00176 | U.227: .WORD | <<GET_SYS_PRO1-U.227>-2> | : |
| | | :TPASTYPE | | : |
| 00000000* | 00178 | U.228: .WORD | 28739 | : |
| | | :TPASADDR | | : |
| 00000010 | 0017C | U.229: .LONG | <<SYSTEM_PROT-U.229>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 00180 | U.230: .LONG | 16 | : |
| | | :TPASTARGET | | : |
| 15F6 | 00182 | U.231: .WORD | <<GET_SYS_PRO1-U.231>-2> | : |
| | | :TPASTYPE | | : |
| 0000* | 00184 | U.232: .WORD | 5622 | : |
| | | :TPASTARGET | | : |
| | | U.233: .WORD | <<U.210-U.233>-2> | : |
| | 00186 | :GET_OWN_PRO | | : |
| | | U.190: .BLKB | 0 | : |
| 003A | 00186 | :TPASTYPE | | : |
| | | U.234: .WORD | 58 | : |
| 003D | 00188 | :TPASTYPE | | : |
| | | U.235: .WORD | 61 | : |
| 15F6 | 0018A | :TPASTYPE | | : |
| | | U.236: .WORD | 5622 | : |
| 0000* | 0018C | :TPASTARGET | | : |
| | | U.237: .WORD | <<U.210-U.237>-2> | : |
| | 0018E | :GET_OWN_PRO1: | | : |
| | | .BLKB | 0 | : |
| 7052 | 0018E | :TPASTYPE | | : |
| | | U.238: .WORD | 28754 | : |
| 00000000* | 00190 | :TPASADDR | | : |
| | | U.239: .LONG | <<OWNER_PROT-U.239>-4> | : |
| 00000001 | 00194 | :TPASMASK | | : |
| | | U.240: .LONG | 1 | : |
| 0000* | 00198 | :TPASTARGET | | : |
| | | U.241: .WORD | <<GET_OWN_PRO1-U.241>-2> | : |
| 7057 | 0019A | :TPASTYPE | | : |
| | | U.242: .WORD | 28759 | : |

| | | | | | | |
|-----------|-------|----------------|------|--------------------------|--|---|
| 00000000* | 0019C | :TPASADDR | | | | |
| | | U.243: | LONG | <<OWNER_PROT-U.243>-4> | | : |
| 00000002 | 001A0 | :TPASMASK | | | | : |
| | | U.244: | LONG | 2 | | : |
| 0000* | 001A4 | :TPASTARGET | | | | : |
| | | U.245: | WORD | <<GET_OWN_PRO1-U.245>-2> | | : |
| 7045 | 001A6 | :TPASTYPE | | | | : |
| | | U.246: | WORD | 28741 | | : |
| 00000000* | 001A8 | :TPASADDR | | | | : |
| | | U.247: | LONG | <<OWNER_PROT-U.247>-4> | | : |
| 00000004 | 001AC | :TPASMASK | | | | : |
| | | U.248: | LONG | 4 | | : |
| 0000* | 001B0 | :TPASTARGET | | | | : |
| | | U.249: | WORD | <<GET_OWN_PRO1-U.249>-2> | | : |
| 7044 | 001B2 | :TPASTYPE | | | | : |
| | | U.250: | WORD | 28740 | | : |
| 00000000* | 001B4 | :TPASADDR | | | | : |
| | | U.251: | LONG | <<OWNER_PROT-U.251>-4> | | : |
| 00000008 | 001B8 | :TPASMASK | | | | : |
| | | U.252: | LONG | 8 | | : |
| 0000* | 001BC | :TPASTARGET | | | | : |
| | | U.253: | WORD | <<GET_OWN_PRO1-U.253>-2> | | : |
| 7043 | 001BE | :TPASTYPE | | | | : |
| | | U.254: | WORD | 28739 | | : |
| 00000000* | 001C0 | :TPASADDR | | | | : |
| | | U.255: | LONG | <<OWNER_PROT-U.255>-4> | | : |
| 00000010 | 001C4 | :TPASMASK | | | | : |
| | | U.256: | LONG | 16 | | : |
| 0000* | 001C8 | :TPASTARGET | | | | : |
| | | U.257: | WORD | <<GET_OWN_PRO1-U.257>-2> | | : |
| 15F6 | 001CA | :TPASTYPE | | | | : |
| | | U.258: | WORD | 5622 | | : |
| 0000* | 001CC | :TPASTARGET | | | | : |
| | | U.259: | WORD | <<U.210-U.259>-2> | | : |
| | 001CE | :GET_GRP_PRO | | | | : |
| | | U.196: | BLKB | 0 | | : |
| 003A | 001CE | :TPASTYPE | | | | : |
| | | U.260: | WORD | 58 | | : |
| 003D | 001D0 | :TPASTYPE | | | | : |
| | | U.261: | WORD | 61 | | : |
| 15F6 | 001D2 | :TPASTYPE | | | | : |
| | | U.262: | WORD | 5622 | | : |
| 0000* | 001D4 | :TPASTARGET | | | | : |
| | | U.263: | WORD | <<U.210-U.263>-2> | | : |
| | 001D6 | :GET_GRP_PRO1: | | | | : |
| | | BLKB | | 0 | | : |
| 7052 | 001D6 | :TPASTYPE | | | | : |
| | | U.264: | WORD | 28754 | | : |
| 00000000* | 001D8 | :TPASADDR | | | | : |
| | | U.265: | LONG | <<GROUP_PROT-U.265>-4> | | : |
| 00000001 | 001DC | :TPASMASK | | | | : |
| | | U.266: | LONG | 1 | | : |
| 0000* | 001E0 | :TPASTARGET | | | | : |
| | | U.267: | WORD | <<GET_GRP_PRO1-U.267>-2> | | : |
| 7057 | 001E2 | :TPASTYPE | | | | : |
| | | U.268: | WORD | 28759 | | : |
| 00000000* | 001E4 | :TPASADDR | | | | : |

| | | | | |
|-----------|-------|----------------|--------------------------|---|
| 00000002 | 001E8 | U.269: .LONG | <<GROUP_PROT-U.269>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 001EC | U.270: .LONG | 2 | : |
| | | :TPASTARGET | | : |
| 7045 | 001EE | U.271: .WORD | <<GET_GRP_PRO1-U.271>-2> | : |
| | | :TPASTYPE | | : |
| 00000000* | 001F0 | U.272: .WORD | 28741 | : |
| | | :TPASADDR | | : |
| 00000004 | 001F4 | U.273: .LONG | <<GROUP_PROT-U.273>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 001F8 | U.274: .LONG | 4 | : |
| | | :TPASTARGET | | : |
| 7044 | 001FA | U.275: .WORD | <<GET_GRP_PRO1-U.275>-2> | : |
| | | :TPASTYPE | | : |
| 00000000* | 001FC | U.276: .WORD | 28740 | : |
| | | :TPASADDR | | : |
| 00000008 | 00200 | U.277: .LONG | <<GROUP_PROT-U.277>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 00204 | U.278: .LONG | 8 | : |
| | | :TPASTARGET | | : |
| 7043 | 00206 | U.279: .WORD | <<GET_GRP_PRO1-U.279>-2> | : |
| | | :TPASTYPE | | : |
| 00000000* | 00208 | U.280: .WORD | 28739 | : |
| | | :TPASADDR | | : |
| 00000010 | 0020C | U.281: .LONG | <<GROUP_PROT-U.281>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 00210 | U.282: .LONG | 16 | : |
| | | :TPASTARGET | | : |
| 15F6 | 00212 | U.283: .WORD | <<GET_GRP_PRO1-U.283>-2> | : |
| | | :TPASTYPE | | : |
| 0000* | 00214 | U.284: .WORD | 5622 | : |
| | | :TPASTARGET | | : |
| | 00216 | U.285: .WORD | <<U.210-U.285>-2> | : |
| | | :GET_WOR_PRO | | : |
| 003A | 00216 | U.202: .BLKB | 0 | : |
| | | :TPASTYPE | | : |
| 003D | 00218 | U.286: .WORD | 58 | : |
| | | :TPASTYPE | | : |
| 15F6 | 0021A | U.287: .WORD | 61 | : |
| | | :TPASTYPE | | : |
| 0000* | 0021C | U.288: .WORD | 5622 | : |
| | | :TPASTARGET | | : |
| | 0021E | U.289: .WORD | <<U.210-U.289>-2> | : |
| | | :GET_WOR_PRO1: | | : |
| 7052 | 0021E | U.202: .BLKB | 0 | : |
| | | :TPASTYPE | | : |
| 00000000* | 00220 | U.290: .WORD | 28754 | : |
| | | :TPASADDR | | : |
| 00000001 | 00224 | U.291: .LONG | <<WORLD_PROT-U.291>-4> | : |
| | | :TPASMASK | | : |
| 0000* | 00228 | U.292: .LONG | 1 | : |
| | | :TPASTARGET | | : |
| 7057 | 0022A | U.293: .WORD | <<GET_WOR_PRO1-U.293>-2> | : |
| | | :TPASTYPE | | : |
| 00000000* | 0022C | U.294: .WORD | 28759 | : |
| | | :TPASADDR | | : |
| | | U.295: .LONG | <<WORLD_PROT-U.295>-4> | : |

| | | | | |
|-----------|-------|--------------|------|----------------------------|
| 00000002 | 00230 | :TPASMASK | | |
| | | U.296: | LONG | 2 |
| 0000* | 00234 | :TPASTARGET | | |
| | | U.297: | WORD | <<GET_WOR_PRO1-U.297>-2> |
| 7045 | 00236 | :TPASTYPE | | |
| | | U.298: | WORD | 28741 |
| 00000000* | 00238 | :TPASADDR | | |
| | | U.299: | LONG | <<WORLD_PROT-U.299>-4> |
| 00000004 | 0023C | :TPASMASK | | |
| | | U.300: | LONG | 4 |
| 0000* | 00240 | :TPASTARGET | | |
| | | U.301: | WORD | <<GET_WOR_PRO1-U.301>-2> |
| 7044 | 00242 | :TPASTYPE | | |
| | | U.302: | WORD | 28740 |
| 00000000* | 00244 | :TPASADDR | | |
| | | U.303: | LONG | <<WORLD_PROT-U.303>-4> |
| 00000008 | 00248 | :TPASMASK | | |
| | | U.304: | LONG | 8 |
| 0000* | 0024C | :TPASTARGET | | |
| | | U.305: | WORD | <<GET_WOR_PRO1-U.305>-2> |
| 7043 | 0024E | :TPASTYPE | | |
| | | U.306: | WORD | 28739 |
| 00000000* | 00250 | :TPASADDR | | |
| | | U.307: | LONG | <<WORLD_PROT-U.307>-4> |
| 00000010 | 00254 | :TPASMASK | | |
| | | U.308: | LONG | 16 |
| 0000* | 00258 | :TPASTARGET | | |
| | | U.309: | WORD | <<GET_WOR_PRO1-U.309>-2> |
| 15F6 | 0025A | :TPASTYPE | | |
| | | U.310: | WORD | 5622 |
| 0000* | 0025C | :TPASTARGET | | |
| | | U.311: | WORD | <<U.210-U.311>-2> |
| | 0025E | :CHK_END_PRO | | |
| | | U.210: | BLKB | 0 |
| 102C | 0025E | :TPASTYPE | | |
| | | U.312: | WORD | 4140 |
| 0000* | 00260 | :TPASTARGET | | |
| | | U.313: | WORD | <<GET_PROT_CLASS-U.313>-2> |
| 1429 | 00262 | :TPASTYPE | | |
| | | U.314: | WORD | 5161 |
| 0000* | 00264 | :TPASTARGET | | |
| | | U.315: | WORD | <<U.73-U.315>-2> |
| | 00266 | :GET_STRING | | |
| | | U.91: | BLKB | 0 |
| 102C | 00266 | :TPASTYPE | | |
| | | U.316: | WORD | 4140 |
| FFFE | 00268 | :TPASTARGET | | |
| | | U.317: | WORD | -2 |
| 1029 | 0026A | :TPASTYPE | | |
| | | U.318: | WORD | 4137 |
| FFFE | 0026C | :TPASTARGET | | |
| | | U.319: | WORD | -2 |
| 11F7 | 0026E | :TPASTYPE | | |
| | | U.320: | WORD | 4599 |
| FFFE | 00270 | :TPASTARGET | | |
| | | U.321: | WORD | -2 |
| 0DF8 | 00272 | :TPASTYPE | | |

```
0000* 00274 U.322: WORD 3576 ;
          :TPASSUBEXP ;
          U.324: WORD <<U.323-U.324>-2> ;
          00276 :GET STRING1 ;
          U.323: BLKB 0 ;
19F8 00276 :TPASTYPE ;
          U.325: WORD 6648 ;
0000* 00278 :TPASSUBEXP ;
          U.327: WORD <<U.326-U.327>-2> ;
0000* 0027A :TPASTARGET ;
          U.328: WORD <<U.323-U.328>-2> ;
15F6 0027C :TPASTYPE ;
          U.329: WORD 5622 ;
FFFF 0027E :TPASTARGET ;
          U.330: WORD -1 ;
          00280 :CHK_DELIM ;
          U.326: BLKB 0 ;
102B 00280 :TPASTYPE ;
          U.331: WORD 4139 ;
FFFE 00282 :TPASTARGET ;
          U.332: WORD -2 ;
102C 00284 :TPASTYPE ;
          U.333: WORD 4140 ;
FFFE 00286 :TPASTARGET ;
          U.334: WORD -2 ;
1029 00288 :TPASTYPE ;
          U.335: WORD 4137 ;
FFFE 0028A :TPASTARGET ;
          U.336: WORD -2 ;
11F7 0028C :TPASTYPE ;
          U.337: WORD 4599 ;
FFFE 0028E :TPASTARGET ;
          U.338: WORD -2 ;
15ED 00290 :TPASTYPE ;
          U.339: WORD 5613 ;
FFFF 00292 :TPASTARGET ;
          U.340: WORD -1 ;
          00294 :CHK_FOR_END ;
          U.73: BLKB 0 ;
15F7 00294 :TPASTYPE ;
          U.341: WORD 5623 ;
FFFF 00296 :TPASTARGET ;
          U.342: WORD -1 ;
```

.PSECT _LIB\$KEY0\$,NOWRT, SHR, PIC,1

```
00000 ACE_KEY::
          00000 :TPASKEY0 BLKB 0
          U.1: BLKB 0
0000* 00000 :TPASKEY
          U.4: WORD <U.3-U.1> ;
0000* 00002 :TPASKEY
          U.12: WORD <U.11-U.1> ;
0000* 00004 :TPASKEY
          U.20: WORD <U.19-U.1> ;
0000* 00006 :TPASKEY
```

```
0000* 00008 U.27: .WORD <U.26-U.1>
:TPASKEY
0000* 0000A U.34: .WORD <U.33-U.1>
:TPASKEY
0000* 0000C U.42: .WORD <U.41-U.1>
:TPASKEY
0000* 0000E U.50: .WORD <U.49-U.1>
:TPASKEY
0000* 00010 U.56: .WORD <U.55-U.1>
:TPASKEY
0000* 00012 U.62: .WORD <U.61-U.1>
:TPASKEY
0000* 00014 U.117: .WORD <U.116-U.1>
:TPASKEY
0000* 00016 U.123: .WORD <U.122-U.1>
:TPASKEY
0000* 00018 U.129: .WORD <U.128-U.1>
:TPASKEY
0000* 0001A U.145: .WORD <U.144-U.1>
:TPASKEY
0000* 0001C U.151: .WORD <U.150-U.1>
:TPASKEY
0000* 0001E U.157: .WORD <U.156-U.1>
:TPASKEY
0000* 00020 U.163: .WORD <U.162-U.1>
:TPASKEY
0000* 00022 U.169: .WORD <U.168-U.1>
:TPASKEY
0000* 00024 U.181: .WORD <U.180-U.1>
:TPASKEY
0000* 00026 U.187: .WORD <U.186-U.1>
:TPASKEY
0000* 00028 U.193: .WORD <U.192-U.1>
:TPASKEY
U.199: .WORD <U.198-U.1>
```

.PSECT \$SPLITS,NOWRT,NOEXE,2

```
44 41 45 52 00000 P.AAC: .ASCII \READ\
00000004 00004 P.AAB: .LONG 4
00000000 00008 P.AAC: .ADDRESS P.AAC
45 54 49 52 57 0000C P.AAE: .ASCII \WRITE\
00011 .BLKB 3
00000005 00014 P.AAD: .LONG 5
00000000 00018 P.AAE: .ADDRESS P.AAE
45 54 55 43 45 58 45 0001C P.AAG: .ASCII \EXECUTE\
00023 .BLKB 1
00000007 00024 P.AAF: .LONG 7
00000000 00028 P.AAG: .ADDRESS P.AAG
45 54 45 4C 45 44 0002C P.AAI: .ASCII \DELETE\
00032 .BLKB 2
00000006 00034 P.AAH: .LONG 6
00000000 00038 P.AAI: .ADDRESS P.AAI
4C 4F 52 54 4E 4F 43 0003C P.AAK: .ASCII \CONTROL\
00043 .BLKB 1
00000007 00044 P.AAJ: .LONG 7
00000000 00048 P.AAK: .ADDRESS P.AAK
```


| | | | | | | | | | | |
|----|----|----|----|----|-------|--------|----------|---------|----------|---|
| 35 | SF | 54 | 49 | 42 | 0004C | P.AAM: | .ASCII | \BIT_5\ | : | |
| | | | | | 00051 | | .BLKB | 3 | : | |
| | | | | | 00054 | P.AAL: | .LONG | 5 | : | |
| | | | | | 00058 | | .ADDRESS | P.AAM | : | |
| 36 | SF | 54 | 49 | 42 | 0005C | P.AAO: | .ASCII | \BIT_6\ | : | |
| | | | | | 00061 | | .BLKB | 3 | : | |
| | | | | | 00064 | P.AAN: | .LONG | 5 | : | |
| | | | | | 00068 | | .ADDRESS | P.AAO | : | |
| 37 | SF | 54 | 49 | 42 | 0006C | P.AAQ: | .ASCII | \BIT_7\ | : | |
| | | | | | 00071 | | .BLKB | 3 | : | |
| | | | | | 00074 | P.AAP: | .LONG | 5 | : | |
| | | | | | 00078 | | .ADDRESS | P.AAQ | : | |
| 38 | SF | 54 | 49 | 42 | 0007C | P.AAS: | .ASCII | \BIT_8\ | : | |
| | | | | | 00081 | | .BLKB | 3 | : | |
| | | | | | 00084 | P.AAR: | .LONG | 5 | : | |
| | | | | | 00088 | | .ADDRESS | P.AAS | : | |
| 39 | SF | 54 | 49 | 42 | 0008C | P.AAU: | .ASCII | \BIT_9\ | : | |
| | | | | | 00091 | | .BLKB | 3 | : | |
| | | | | | 00094 | P.AAT: | .LONG | 5 | : | |
| | | | | | 00098 | | .ADDRESS | P.AAU | : | |
| 30 | 31 | SF | 54 | 49 | 42 | 0009C | P.AAW: | .ASCII | \BIT_10\ | : |
| | | | | | 000A2 | | .BLKB | 2 | : | |
| | | | | | 000A4 | P.AAV: | .LONG | 6 | : | |
| | | | | | 000A8 | | .ADDRESS | P.AAW | : | |
| 31 | 31 | SF | 54 | 49 | 42 | 000AC | P.AAY: | .ASCII | \BIT_11\ | : |
| | | | | | 000B2 | | .BLKB | 2 | : | |
| | | | | | 000B4 | P.AAX: | .LONG | 6 | : | |
| | | | | | 000B8 | | .ADDRESS | P.AAY | : | |
| 32 | 31 | SF | 54 | 49 | 42 | 000BC | P.ABA: | .ASCII | \BIT_12\ | : |
| | | | | | 000C2 | | .BLKB | 2 | : | |
| | | | | | 000C4 | P.AAZ: | .LONG | 6 | : | |
| | | | | | 000C8 | | .ADDRESS | P.ABA | : | |
| 33 | 31 | SF | 54 | 49 | 42 | 000CC | P.ABC: | .ASCII | \BIT_13\ | : |
| | | | | | 000D2 | | .BLKB | 2 | : | |
| | | | | | 000D4 | P.ABB: | .LONG | 6 | : | |
| | | | | | 000D8 | | .ADDRESS | P.ABC | : | |
| 34 | 31 | SF | 54 | 49 | 42 | 000DC | P.ABE: | .ASCII | \BIT_14\ | : |
| | | | | | 000E2 | | .BLKB | 2 | : | |
| | | | | | 000E4 | P.ABD: | .LONG | 6 | : | |
| | | | | | 000E8 | | .ADDRESS | P.ABE | : | |
| 35 | 31 | SF | 54 | 49 | 42 | 000EC | P.ABG: | .ASCII | \BIT_15\ | : |
| | | | | | 000F2 | | .BLKB | 2 | : | |
| | | | | | 000F4 | P.ABF: | .LONG | 6 | : | |
| | | | | | 000F8 | | .ADDRESS | P.ABG | : | |
| 36 | 31 | SF | 54 | 49 | 42 | 000FC | P.ABI: | .ASCII | \BIT_16\ | : |
| | | | | | 00102 | | .BLKB | 2 | : | |
| | | | | | 00104 | P.ABH: | .LONG | 6 | : | |
| | | | | | 00108 | | .ADDRESS | P.ABI | : | |
| 37 | 31 | SF | 54 | 49 | 42 | 0010C | P.ABK: | .ASCII | \BIT_17\ | : |
| | | | | | 00112 | | .BLKB | 2 | : | |
| | | | | | 00114 | P.ABJ: | .LONG | 6 | : | |
| | | | | | 00118 | | .ADDRESS | P.ABK | : | |
| 38 | 31 | SF | 54 | 49 | 42 | 0011C | P.ABM: | .ASCII | \BIT_18\ | : |
| | | | | | 00122 | | .BLKB | 2 | : | |
| | | | | | 00124 | P.ABL: | .LONG | 6 | : | |
| | | | | | 00128 | | .ADDRESS | P.ABM | : | |
| 39 | 31 | SF | 54 | 49 | 42 | 0012C | P.ABO: | .ASCII | \BIT_19\ | : |

```
00000006 00132
00000000 00134 P.ABN: .BLKB 2
30 32 5F 54 49 42 00138 .LONG 6
0013C P.ABQ: .ADDRESS P.ABQ
00142 .ASCII \BIT_20\
00000006 00144 P.ABP: .BLKB 2
00000000 00148 .LONG 6
31 32 5F 54 49 42 0014C P.ABS: .ADDRESS P.ABQ
00152 .ASCII \BIT_21\
00154 P.ABR: .BLKB 2
00000006 00158 .LONG 6
00000000 0015C P.ABU: .ADDRESS P.ABS
32 32 5F 54 49 42 00162 .ASCII \BIT_22\
00164 P.ABT: .BLKB 2
00000006 00168 .LONG 6
00000000 0016C P.ABW: .ADDRESS P.ABU
33 32 5F 54 49 42 00172 .ASCII \BIT_23\
00174 P.ABV: .BLKB 2
00000006 00178 .LONG 6
00000000 0017C P.ABY: .ADDRESS P.ABW
34 32 5F 54 49 42 00182 .ASCII \BIT_24\
00184 P.ABX: .BLKB 2
00000006 00188 .LONG 6
00000000 0018C P.ACA: .ADDRESS P.ABY
35 32 5F 54 49 42 00192 .ASCII \BIT_25\
00194 P.ABZ: .BLKB 2
00000006 00198 .LONG 6
00000000 0019C P.ACC: .ADDRESS P.ACA
36 32 5F 54 49 42 001A2 .ASCII \BIT_26\
001A4 P.ACB: .BLKB 2
00000006 001A8 .LONG 6
00000000 001AC P.ACE: .ADDRESS P.ACC
37 32 5F 54 49 42 001B2 .ASCII \BIT_27\
001B4 P.ACD: .BLKB 2
00000006 001B8 .LONG 6
00000000 001BC P.ACG: .ADDRESS P.ACE
38 32 5F 54 49 42 001C2 .ASCII \BIT_28\
001C4 P.ACF: .BLKB 2
00000006 001C8 .LONG 6
00000000 001CC P.ACI: .ADDRESS P.ACG
39 32 5F 54 49 42 001D2 .ASCII \BIT_29\
001D4 P.ACH: .BLKB 2
00000006 001D8 .LONG 6
00000000 001DC P.ACK: .ADDRESS P.ACI
30 33 5F 54 49 42 001E2 .ASCII \BIT_30\
001E4 P.ACJ: .BLKB 2
00000006 001E8 .LONG 6
00000000 001EC P.ACM: .ADDRESS P.ACK
31 33 5F 54 49 42 001F2 .ASCII \BIT_31\
001F4 P.ACL: .BLKB 2
00000006 001F8 .LONG 6
00000000 001FC P.AAA: .ADDRESS P.ACM
00000000 00214 P.AAB, P.AAD, P.AAF, P.AAH, P.AAJ, -
00000000 0022C P.AAL, P.AAN, P.AAP, P.AAR, P.AAT, P.AAV, -
00000000 00244 P.AAX, P.AAZ, P.ABB, P.ABD, P.ABF, P.ABH, -
00000000 0025C P.ABJ, P.ABL, P.ABN, P.ABP, P.ABR, P.ABT, -
00000000 00274 P.ABV, P.ABX, P.ABZ, P.ACB, P.ACD, P.ACF, -
P.ACH, P.ACJ, P.ACL
```

```
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000
```

```
4B 43 4F 4C 24 4C 43 41 0027C P.ACP: .ASCII \ACL$LOCK\
00000008' 00284 P.ACO: .LONG 8
00000000' 00288 .ADDRESS P.ACP
5F 4C 49 46 24 4C 43 41 0028C P.ACR: .ASCII \ACL$FIL_\
00000008' 00294 P.ACQ: .LONG 8
00000000' 00298 .ADDRESS P.ACR
5F 56 45 44 24 4C 43 41 0029C P.ACT: .ASCII \ACL$DEV_\
00000008' 002A4 P.ACS: .LONG 8
000000C0' 002A8 .ADDRESS P.ACT
5F 43 42 4A 24 4C 43 41 002AC P.ACV: .ASCII \ACL$JBC_\
00000008' 002B4 P.ACU: .LONG 8
00000000' 002B8 .ADDRESS P.ACV
5F 46 45 43 24 4C 43 41 002BC P.ACX: .ASCII \ACL$CEF_\
00000008' 002C4 P.ACW: .LONG 8
00000000' 002C8 .ADDRESS P.ACX
5F 54 4E 4C 24 4C 43 41 002CC P.ACZ: .ASCII \ACL$LNT_\
00000008' 002D4 P.ACY: .LONG 8
00000000' 002D8 .ADDRESS P.ACZ
5F 43 52 50 24 4C 43 41 002DC P.ADB: .ASCII \ACL$PRC_\
00000008' 002E4 P.ADA: .LONG 8
00000000' 002E8 .ADDRESS P.ADB
5F 4C 42 47 24 4C 43 41 002EC P.ADD: .ASCII \ACL$GBL_\
00000008' 002F4 P.ADC: .LONG 8
00000000' 002F8 .ADDRESS P.ADD
00000000' 002FC P.ACN: .ADDRESS P.ACO, P.ACQ, P.ACS, P.ACU, P.ACW, -
00000000' 00314 P.ACY, P.ADA, P.ADC
59 54 49 52 55 43 45 53 0031C P.ADE: .ASCII \SECURITY\
```

.PSECT \$OWNS,NOEXE,2

```
00 00000 JOURNAL_ACES:
      .BYTE 0
00001 .BLKB 3
00004 ACE_BUFFER:
      .BLKB 512
00204 ACE_INDEX:
      .BLKB 4
00208 ACE_TYPE:
      .BLKB 4
0020C ACE_RIGHTS:
      .BLKB 4
00210 UIC_FLAGS:
      .BLKB 4
00214 UIC_COUNT:
      .BLKB 4
00218 IDENTIFIER:
      .BLKB 4
0021C ID_NAME: .BLKB 8
00224 ID_COUNT:
      .BLKB 4
00228 JOURNAL_NAME:
      .BLKB 8
00230 ACCESS_FLAGS:
      .BLKB 4
00234 SYSTEM_PROT:
      .BLKB 4
00238 OWNER_PROT:
```


| | | | |
|-------|-----------------|-------|-----|
| 0023C | GROUP_PROT: | .BLKB | 4 |
| 00240 | WORLD_PROT: | .BLKB | 4 |
| 00244 | BIT_NAME_TABLE: | .BLKB | 4 |
| 00248 | CHANGE_ACMODE: | .BLKB | 4 |
| 0024C | CALL_ACMODE: | .BLKB | 4 |
| 00250 | PARENT_ID: | .BLKB | 4 |
| 00254 | ACL_QUEUE_HEAD: | .BLKB | 4 |
| 00258 | ACL_POINTER: | .BLKB | 4 |
| 0025C | ACL_SPLIT: | .BLKB | 4 |
| 00260 | ACE_POINTER: | .BLKB | 4 |
| 00264 | ACE_NUMBER: | .BLKB | 4 |
| 00268 | ACL_AREA: | .BLKB | 512 |
| 00468 | ACL_CONTEXT: | .BLKB | 4 |
| 0046C | LOCK_RESNAM: | .BLKB | 8 |
| 00474 | RESNAM_TEXT: | .BLKB | 31 |

```

DEFAULT_BITS=
LOCK_PREFIX=

```

P.AAA
P.ACN

```

. EXTRN  ACL_ADDENTRY, ACL_DELENTTRY
. EXTRN  ACL_MODENTRY, ACL_FINDENTRY
. EXTRN  ACL_FINDTYPE, ACL_DELETEACL
. EXTRN  ACL_READACL, ACL_ACLLENGTH
. EXTRN  ACL_READACE, ACL_LOCATEACE
. EXTRN  ACL_INIT_QUEUE, ALLOC_PAGED
. EXTRN  DALLOC_PAGED, LIB$TPARSE
. EXTRN  LIB$FID_TO_NAME
. EXTRN  LIB$GET_VM, LIB$FREE_VM
. EXTRN  EXESPROBER, EXESPROBEW
. EXTRN  IOC$VERIFYCHAN, SCH$LOCKR
. EXTRN  SCH$LOCKW, SCH$UNLOCK
. EXTRN  CTLSGL_PCB, LIB$SYNTAXERR

```

.PSECT SCODES,NOWRT,2

```

.ENTRY    SYSSPARSE ACL, Save R2,R3,R4,R5,R6
MOVAB     JOURNAL_NAME, R6
SUBL2     #36, SP-
CLRL      BIT_NAME_TABLE
MOVL      BIT_TABLE, R0
BEQL      1$
PROBER    #0, #256, (R0)

```

0616
0676
0677
0678

| | | | | | | |
|----|------|----|-----------|----|------|-------|
| | | | | | 007C | 00000 |
| | | 56 | 00000000' | EF | 9E | 00002 |
| | | 5E | | 24 | C2 | 00009 |
| | | | 1C | A6 | D4 | 0000C |
| | | 50 | 10 | AC | D0 | 0000F |
| | | | | 0C | 13 | 00013 |
| 60 | 0100 | 8F | | 00 | 0C | 00015 |

| | | | | | | | | | | | |
|------|----|-----------|----|-----------|------|----|-------|--------|--------------------------------------|--------------------------------|------|
| 0200 | 8F | 00 | 1C | A6 | 5B | 13 | 0001B | BEQL | 2\$ | | |
| | | | | 6E | 50 | D0 | 0001D | MOVL | R0, BIT_NAME_TABLE | 0679 | |
| | | | | | 00 | 2C | 00021 | MOVCS | #0, (SPT), #0, #512, ACE_BUFFER | 0684 | |
| | | | | | FDDC | C6 | 00028 | | | | |
| | | | | | E0 | A6 | 7C | 0002B | CLRQ | ACE_TYPE | 0685 |
| | | | | | DC | A6 | D4 | 0002E | CLRL | ACE_INDEX | |
| | | | | | E8 | A6 | 7C | 00031 | CLRQ | UIC_FLAGS | 0686 |
| | | | | | FC | A6 | D4 | 00034 | CLRL | ID_COUNT | 0687 |
| | | | | | F4 | A6 | B4 | 00037 | CLRW | ID_NAME | 0689 |
| | | | | | | 66 | B4 | 0003A | CLRW | JOURNAL_NAME | |
| | | | | | 14 | A6 | 7C | 0003C | CLRQ | GROUP_PROT | 0690 |
| | | | | | 10 | A6 | D4 | 0003F | CLRL | OWNER_PROT | |
| | | | | | 08 | A6 | 7C | 00042 | CLRQ | ACCESS_FLAGS | 0688 |
| 24 | | 00 | | 6E | 00 | 2C | 00045 | MOVCS | #0, (SP), #0, #36, TPARSE_BLOCK | 0692 | |
| | | | | | 6E | | 0004A | | | | |
| | | | | 6E | 08 | D0 | 0004B | MOVL | #8, TPARSE_BLOCK | 0693 | |
| | | 04 | | AE | 02 | 88 | 0004E | BISB2 | #2, TPARSE_BLOCK+4 | 0694 | |
| | | | | 54 | 04 | AC | D0 | 00052 | MOVL | ACL_STRING, R4 | 0695 |
| | | 64 | | 08 | 00 | 0C | 00056 | PROBER | #0, #8, (R4) | | |
| | | | | | 1C | 13 | 0005A | BEQL | 2\$ | | |
| | | | | 50 | 64 | 3C | 0005C | MOVZWL | (R4), R0 | 0698 | |
| | | 08 | | AE | 50 | D0 | 0005F | MOVL | R0, TPARSE_BLOCK+8 | | |
| | | | | 55 | 50 | D0 | 00063 | MOVL | R0, ACL_STRING_LEN | | |
| | | | | 50 | 04 | A4 | D0 | 00066 | MOVL | 4(R4), R0 | 0699 |
| | | | | 51 | 55 | D0 | 0006A | MOVL | ACL_STRING_LEN, R1 | | |
| | | | | | 53 | D4 | 0006D | CLRL | R3 | | |
| | | | | 00000000G | 00 | 16 | 0006F | JSB | EXESPROBER | | |
| | | 03 | | | 50 | E8 | 00075 | BLBS | R0, 3\$ | | |
| | | | | 04 | 017E | 31 | 00078 | BRW | 17\$ | | |
| | | 0C | | AE | A4 | D0 | 0007B | MOVL | 4(R4), TPARSE_BLOCK+12 | 0700 | |
| | | | | 00000000' | EF | 9F | 00080 | PUSHAB | ACE_KEY | 0705 | |
| | | | | 00000000' | EF | 9F | 00086 | PUSHAB | ACE_STATE | | |
| | | | | | 08 | AE | 9F | 0008C | PUSHAB | TPARSE_BLOCK | |
| | | 00000000G | | 00 | 03 | FB | 0008F | CALLS | #3, LIB\$TPARSE | | |
| | | | | 51 | 0C | AC | D0 | 00096 | MOVL | ERROR_POSITION, R1 | 0709 |
| | | | | | 08 | 13 | 0009A | BEQL | 4\$ | | |
| 61 | | 02 | | | 00 | 0D | 0009C | PROBEW | #0, #2, (R1) | 0710 | |
| | | | | | D6 | 13 | 000A0 | BEQL | 2\$ | | |
| 61 | | 55 | | 08 | AE | A3 | 000A2 | SUBW3 | TPARSE_BLOCK+8, ACL_STRING_LEN, (R1) | 0712 | |
| | | 00000000G | | 8F | 50 | D1 | 000A7 | CMPL | STATUS, #LIB\$_SYNTAXERR | 0718 | |
| | | | | | 44 | 13 | 000AE | BEQL | 8\$ | | |
| | | | | | E0 | A6 | D5 | 000B0 | TSTL | ACE_TYPE | 0719 |
| | | | | | 3F | 13 | 000B3 | BEQL | 8\$ | | |
| | | 05 | | | 50 | E9 | 000B5 | BLBC | STATUS, 5\$ | 0720 | |
| | | | | | 08 | AE | D5 | 000B8 | TSTL | TPARSE_BLOCK+8 | |
| | | | | | 37 | 14 | 000BB | BGTR | 8\$ | | |
| | | 01 | | | 50 | E8 | 000BD | BLBS | STATUS, 6\$ | 0723 | |
| | | | | | 04 | | 000C0 | RET | | | |
| | | FDDD | | C6 | E0 | A6 | 90 | 000C1 | MOVB | ACE_TYPE, ACE_BUFFER+1 | 0727 |
| | | | | 50 | FDDC | C6 | 3C | 000C7 | MOVZWL | ACE_BUFFER+2, R0 | 0729 |
| | | | | 50 | E8 | A6 | C8 | 000CC | BISL2 | UIC_FLAGS, R0 | |
| | | FDDE | | C6 | 08 | A6 | A9 | 000D0 | BISW3 | ACCESS_FLAGS, R0, ACE_BUFFER+2 | 0730 |
| | | | | | E4 | A6 | D0 | 000D7 | MOVL | ACE_RIGHTS, ACE_BUFFER+4 | 0731 |
| | | FDE0 | | C6 | E0 | A6 | CF | 000DD | CASEL | ACE_TYPE, #1, #8 | 0736 |
| | | | | 01 | | | | | .WORD | 9\$-7\$, - | |
| 0039 | | 0039 | | 0039 | 0015 | | 000E2 | | | 10\$-7\$, - | |
| 009F | | 009F | | 0061 | 0061 | | 000EA | | | 10\$-7\$, - | |
| | | | | | 008C | | 000F2 | | | | |

| Address | Op | Op2 | Op3 | Op4 | Op5 | Op6 | Op7 | Op8 | Op9 | Op10 | Op11 | Op12 | Op13 | Op14 | Op15 | Op16 | Op17 | Op18 | Op19 | Op20 | Op21 | Op22 | Op23 | Op24 | Op25 | Op26 | Op27 | Op28 | Op29 | Op30 | Op31 | Op32 | Op33 | Op34 | Op35 | Op36 | Op37 | Op38 | Op39 | Op40 | Op41 | Op42 | Op43 | Op44 | Op45 | Op46 | Op47 | Op48 | Op49 | Op50 | Op51 | Op52 | Op53 | Op54 | Op55 | Op56 | Op57 | Op58 | Op59 | Op60 | Op61 | Op62 | Op63 | Op64 | Op65 | Op66 | Op67 | Op68 | Op69 | Op70 | Op71 | Op72 | Op73 | Op74 | Op75 | Op76 | Op77 | Op78 | Op79 | Op80 | Op81 | Op82 | Op83 | Op84 | Op85 | Op86 | Op87 | Op88 | Op89 | Op90 | Op91 | Op92 | Op93 | Op94 | Op95 | Op96 | Op97 | Op98 | Op99 | Op100 | Op101 | Op102 | Op103 | Op104 | Op105 | Op106 | Op107 | Op108 | Op109 | Op110 | Op111 | Op112 | Op113 | Op114 | Op115 | Op116 | Op117 | Op118 | Op119 | Op120 | Op121 | Op122 | Op123 | Op124 | Op125 | Op126 | Op127 | Op128 | Op129 | Op130 | Op131 | Op132 | Op133 | Op134 | Op135 | Op136 | Op137 | Op138 | Op139 | Op140 | Op141 | Op142 | Op143 | Op144 | Op145 | Op146 | Op147 | Op148 | Op149 | Op150 | Op151 | Op152 | Op153 | Op154 | Op155 | Op156 | Op157 | Op158 | Op159 | Op160 | Op161 | Op162 | Op163 | Op164 | Op165 | Op166 | Op167 | Op168 | Op169 | Op170 | Op171 | Op172 | Op173 | Op174 | Op175 | Op176 | Op177 | Op178 | Op179 | Op180 | Op181 | Op182 | Op183 | Op184 | Op185 | Op186 | Op187 | Op188 | Op189 | Op190 | Op191 | Op192 | Op193 | Op194 | Op195 | Op196 | Op197 | Op198 | Op199 | Op200 | Op201 | Op202 | Op203 | Op204 | Op205 | Op206 | Op207 | Op208 | Op209 | Op210 | Op211 | Op212 | Op213 | Op214 | Op215 | Op216 | Op217 | Op218 | Op219 | Op220 | Op221 | Op222 | Op223 | Op224 | Op225 | Op226 | Op227 | Op228 | Op229 | Op230 | Op231 | Op232 | Op233 | Op234 | Op235 | Op236 | Op237 | Op238 | Op239 | Op240 | Op241 | Op242 | Op243 | Op244 | Op245 | Op246 | Op247 | Op248 | Op249 | Op250 | Op251 | Op252 | Op253 | Op254 | Op255 | Op256 | Op257 | Op258 | Op259 | Op260 | Op261 | Op262 | Op263 | Op264 | Op265 | Op266 | Op267 | Op268 | Op269 | Op270 | Op271 | Op272 | Op273 | Op274 | Op275 | Op276 | Op277 | Op278 | Op279 | Op280 | Op281 | Op282 | Op283 | Op284 | Op285 | Op286 | Op287 | Op288 | Op289 | Op290 | Op291 | Op292 | Op293 | Op294 | Op295 | Op296 | Op297 | Op298 | Op299 | Op300 | Op301 | Op302 | Op303 | Op304 | Op305 | Op306 | Op307 | Op308 | Op309 | Op310 | Op311 | Op312 | Op313 | Op314 | Op315 | Op316 | Op317 | Op318 | Op319 | Op320 | Op321 | Op322 | Op323 | Op324 | Op325 | Op326 | Op327 | Op328 | Op329 | Op330 | Op331 | Op332 | Op333 | Op334 | Op335 | Op336 | Op337 | Op338 | Op339 | Op340 | Op341 | Op342 | Op343 | Op344 | Op345 | Op346 | Op347 | Op348 | Op349 | Op350 | Op351 | Op352 | Op353 | Op354 | Op355 | Op356 | Op357 | Op358 | Op359 | Op360 | Op361 | Op362 | Op363 | Op364 | Op365 | Op366 | Op367 | Op368 | Op369 | Op370 | Op371 | Op372 | Op373 | Op374 | Op375 | Op376 | Op377 | Op378 | Op379 | Op380 | Op381 | Op382 | Op383 | Op384 | Op385 | Op386 | Op387 | Op388 | Op389 | Op390 | Op391 | Op392 | Op393 | Op394 | Op395 | Op396 | Op397 | Op398 | Op399 | Op400 | Op401 | Op402 | Op403 | Op404 | Op405 | Op406 | Op407 | Op408 | Op409 | Op410 | Op411 | Op412 | Op413 | Op414 | Op415 | Op416 | Op417 | Op418 | Op419 | Op420 | Op421 | Op422 | Op423 | Op424 | Op425 | Op426 | Op427 | Op428 | Op429 | Op430 | Op431 | Op432 | Op433 | Op434 | Op435 | Op436 | Op437 | Op438 | Op439 | Op440 | Op441 | Op442 | Op443 | Op444 | Op445 | Op446 | Op447 | Op448 | Op449 | Op450 | Op451 | Op452 | Op453 | Op454 | Op455 | Op456 | Op457 | Op458 | Op459 | Op460 | Op461 | Op462 | Op463 | Op464 | Op465 | |
|---------|----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|
|---------|----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--|

SYSACLSRV
V04-000

\$PARSE_ACL system service

F 15

16-Sep-1984 01:51:51

14-Sep-1984 12:40:53

VAX-11 Bliss-32 V4.0-742

[LOADSS.SRC]SYSACLSRV.B32;1

Page 36
(4)

| | | | | | | | | | | | |
|----|------|----|------|----|----|-------|-------|--------|-----------------------------------|---------------------------|------|
| | | | FDE4 | C6 | 0C | A6 | 7D | 0019D | MOVQ | SYSTEM PROT, ACE BUFFER+8 | 0786 |
| | | | 14 | A6 | 14 | A6 | D2 | 001A3 | MCOML | GROUP PROT, GROUP PROT | 0790 |
| 14 | A6 | 1B | 05 | 05 | 00 | F0 | 001A8 | INSV | #0, #5, #27, GROUP PROT | 0791 | |
| | | | 18 | A6 | 18 | A6 | D2 | 001AE | MCOML | WORLD PROT, WORLD PROT | 0793 |
| 18 | A6 | 1B | 05 | 05 | 00 | F0 | 001B3 | INSV | #0, #5, #27, WORLD PROT | 0794 | |
| | | | FDEC | C6 | 14 | A6 | 7D | 001B9 | MOVQ | GROUP PROT, ACE_BUFFER+16 | 0792 |
| | | | FDDC | C6 | 18 | 90 | 001BF | MOVB | #24, ACE_BUFFER | 0796 | |
| | | | | 54 | 08 | AC | D0 | 001C4 | MOVL | ACL_ENTRY, R4 | 0804 |
| | | 64 | | 08 | 00 | 0C | 001C8 | PROBER | #0, #8, (R4) | | |
| | | | | 51 | 2B | 13 | 001CC | BEQL | 17\$ | | |
| | | | | 08 | 64 | 3C | 001CE | MOVZWL | (R4), ACL_ENTRY_LEN | 0807 | |
| 51 | FDDC | C6 | | 08 | 00 | ED | 001D1 | CMPZV | #0, #8, ACE_BUFFER, ACL_ENTRY_LEN | 0808 | |
| | | | | 50 | 1F | 1A | 001D8 | BGTRU | 17\$ | | |
| | | | | 04 | A4 | D0 | 001DA | MOVL | 4(R4), R0 | 0809 | |
| | | | | | 53 | D4 | 001DE | CLRL | R3 | | |
| | | | | | 00 | 16 | 001E0 | JSB | EXESPROBEW | | |
| | | | | 10 | 50 | E9 | 001E6 | BLBC | R0, 17\$ | | |
| | | | | 50 | C6 | 9A | 001E9 | MOVZBL | ACE_BUFFER, R0 | 0810 | |
| 64 | | 00 | FDDC | C6 | 50 | 2C | 001EE | MOVCS | R0, ACE_BUFFER, #0, (R4), @4(R4) | 0811 | |
| | | | | | 04 | B4 | 001F5 | | | | |
| | | | | 50 | 04 | 11 | 001F7 | BRB | 18\$ | 0810 | |
| | | | | | 0C | D0 | 001F9 | MOVL | #12, R0 | 0814 | |
| | | | | 50 | | 04 | 001FC | RET | | | |
| | | | | | 01 | D0 | 001FD | MOVL | #1, R0 | 0816 | |
| | | | | | 04 | 00200 | RET | | | 0818 | |

; Routine Size: 513 bytes, Routine Base: \$CODE\$ + 0000


```
823 0819 1 %SBTTL '$FORMAT_ACL system service'
824 0820 1 GLOBAL ROUTINE SYSS$FORMAT_ACL (ACL_ENTRY, ACL_LENGTH, ACL_STRING,
825 0821 1 LINE_WIDTH, TERM_DESC, LINE_INDENT,
826 0822 1 BIT_TABLE) =
827 0823 1
828 0824 1 ++
829 0825 1
830 0826 1 FUNCTIONAL DESCRIPTION:
831 0827 1
832 0828 1 This routine converts the Access Control Entry from a binary form
833 0829 1 to a text form.
834 0830 1
835 0831 1 CALLING SEQUENCE:
836 0832 1 SYSS$FORMAT_ACL (ARG1, ARG2, ARG3, ARG4, ARG5, ARG6, ARG7)
837 0833 1
838 0834 1 INPUT PARAMETERS:
839 0835 1 ARG1: address of the input buffer descriptor
840 0836 1 ARG4: address of the maximum line width for formatting
841 0837 1 ARG5: address of the output line segment terminator descriptor
842 0838 1 ARG6: address of the number of columns to indent each line segment
843 0839 1 ARG7: address of an access bit name table
844 0840 1
845 0841 1 IMPLICIT INPUTS:
846 0842 1 none
847 0843 1
848 0844 1 OUTPUT PARAMETERS:
849 0845 1 ARG2: address of a word to get the length of the formatted ACE
850 0846 1 ARG3: address of the output text buffer descriptor
851 0847 1
852 0848 1 IMPLICIT OUTPUTS:
853 0849 1 none
854 0850 1
855 0851 1 ROUTINE VALUE:
856 0852 1 SSS_NORMAL: The conversion was successful.
857 0853 1 SSS_NOSUCHID: The identifier specified in the ACE is not in the
858 0854 1 rights database.
859 0855 1 SSS_BUFFEROVF: The conversion was successful. The formatted ACE
860 0856 1 has overflowed the output buffer and has been
861 0857 1 truncated.
862 0858 1
863 0859 1 SIDE EFFECTS:
864 0860 1 none
865 0861 1
866 0862 1 --
867 0863 1
868 0864 2 BEGIN
869 0865 2
870 M 0866 2 MACRO CHECK_WIDTH (TEST_WIDTH) =
871 M 0867 2 BEGIN
872 M 0868 2 IF .WIDTH GTRU 0
873 M 0869 2 AND .LINE_SIZE + TEST_WIDTH GTRU .WIDTH
874 M 0870 2 THEN
875 M 0871 2 BEGIN
876 M 0872 2 IF .TERM_LENGTH GTR 0
877 M 0873 2 THEN
878 M 0874 2 BEGIN
879 M 0875 2 CHSMOVE (.TERM_LENGTH, .TERM_POINTER, BUFFER[.SIZE]);
```

```
880      SIZE = .SIZE + .TERM_DESC[DSCSW_LENGTH];
881      END;
882      CH$FILL ('C' ' ', .INDENT, BUFFER[.SIZE]);
883      SIZE = .SIZE + .INDENT;
884      LINE_SIZE = .INDENT;
885      END;
886      LINE_SIZE = .LINE_SIZE + TEST_WIDTH;
887      END
888      X,
889
890      STORE_TEXT (STRING) =
891      BEGIN
892      CHECK_WIDTH (XCHARCOUNT (STRING));
893      CH$MOVE (XCHARCOUNT (STRING), UPLIT (STRING), BUFFER[.SIZE]);
894      SIZE = .SIZE + XCHARCOUNT (STRING);
895      END
896      X,
897
898      NEW_LINE =
899      BEGIN
900      IF .TERM_LENGTH GTR 0
901      THEN
902      BEGIN
903      CH$MOVE (.TERM_LENGTH, .TERM_POINTER, BUFFER[.SIZE]);
904      SIZE = .SIZE + .TERM_LENGTH;
905      END;
906      CH$FILL ('C' ' ', .INDENT, BUFFER[.SIZE]);
907      SIZE = .SIZE + .INDENT;
908      LINE_SIZE = .INDENT;
909      END
910      X;
911
912      MAP
913      ACL_ENTRY      : REF $BLOCK,      ! Address of the input descriptor
914      ACL_STRING     : REF $BLOCK,      ! Address of the output descriptor
915      TERM_DESC      : REF $BLOCK;      ! Segment terminator descriptor
916
917      LITERAL
918      MAX_FAO_LENGTH = MAXU (2 * KGB$S_NAME + 3,      ! Max size of
919                           ATR$S_FILE_SPEC),          ! FAO buffer
920      MAX_FMT_ACE    = 3072,                      ! Largest possible formatted ACE
921      VOLNAM_SIZE    = XCHARCOUNT ('DISK$') + ACCESS_VOLNAM + 1; ! Size of full volume name
922
923      LOCAL
924      ACL_ENTRY_LEN, ! Length of input ACE buffer
925      LOCAL_ACE      : $BLOCK [ATR$S_ADDACLENT],      ! Local copy of ACE
926      FAO_DESCR      : $BLOCK [DSC$C_S_BLN],          ! FAO output descriptor
927      KEY_IDENTIFIER : $BLOCK [4],                    ! Key identifier
928      PROT_VALUE     : $BLOCK [4],                    ! Protection value from ACE
929      PROT_FIELD_DSC : REF $BLOCK,                    ! Addr of protection field name
930      PROT_BUF       : VECTOR [32,BYTE],              ! Storage for ASCII protection string
931      PROT_IDX       : VECTOR [32,BYTE],              ! Index into protection string
932      BUFFER         : VECTOR [MAX_FMT_ACE, BYTE],    ! Temp storage for formatted ACE
933
934      LINE_SIZE,     ! Size of the current segment
935      SIZE,          ! Size of formatted ACE
936
```

```
0937      INDENT,                                | Number of columns to indent
0938      WIDTH,                                  | Width of the line
0939      TERM_LENGTH,                             | Size of terminator string
0940      TERM_POINTER,                           | Address of terminator string
0941      FAO_DESC : $BBLOCK [DSC$C_S_BLN],       | FAO output descriptor
0942      FAO_BUF  : VECTOR [MAX_FAO_LENGTH, BYTE], | FAO output buffer
0943      BIT_NAME_DESC : REF $BBLOCK,             | Descr for access bit name
0944      FLAGS      : BITVECTOR [16],             | Flags from ACE
0945      ACCESS_MASK : BYTE,                      | Access mask in ACE
0946      AUDIT_MASK  : BYTE,                      | Audit access mask in ACE
0947      VOLNAM_DESC : $BBLOCK [DSC$C_S_BLN],     | Volume name descriptor
0948      VOLNAM_TEXT  : VECTOR [VOLNAM_SIZE, BYTE], | Volume name storage
0949      FILENAME_DESC : $BBLOCK [DSC$C_S_BLN],   | File name descriptor
0950      FILENAME_TEXT : VECTOR [ATTR$FICE_SPEC], | File name storage
0951      ACL_STRING_LEN,                             | Length of ACL string buffer
0952      LOCAL_STATUS;                               | Local routine return status
0953
0954      ! Protection code names.
0955
0956      BIND
0957          PROT_CODE      = UPLIT BYTE ('R', 'W', 'E', 'D', 'C',
0958                                     REP'27 OF (0)) : VECTOR [, BYTE];
0959
0960      ! Probe the output string buffer.
0961
0962      IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .ACL_STRING)
0963      THEN
0964          BEGIN
0965              ACL_STRING_LEN = .ACL_STRING[DSC$W_LENGTH];
0966              IF EXESPROBER (0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
0967              THEN CH$FILL (%C, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
0968              ELSE RETURN SS$ACCVIO;
0969          END
0970      ELSE RETURN SS$ACCVIO;
0971
0972      ! Set up initial parameters.
0973
0974      INDENT = WIDTH = 0;
0975      TERM_LENGTH = TERM_POINTER = 0;
0976      ACCESS_MASK = AUDIT_MASK = 0;
0977
0978      ! Check the optional arguments.
0979
0980      IF .LINE_WIDTH NEQA 0
0981      THEN IF PROBER (%REF (0), %REF (4), .LINE_WIDTH)
0982      THEN WIDTH = .LINE_WIDTH
0983      ELSE RETURN SS$ACCVIO;
0984
0985      IF .TERM_DESC NEQA 0
0986      THEN
0987          BEGIN
0988              IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .TERM_DESC)
0989              THEN
0990                  BEGIN
0991                      TERM_LENGTH = .TERM_DESC[DSC$W_LENGTH];
0992                      TERM_POINTER = .TERM_DESC[DSC$A_POINTER];
0993                      IF NOT EXESPROBER (0, .TERM_LENGTH, .TERM_POINTER)
```

\$FORMAT_ACL system service

```
0994      0990      4      THEN RETURN SS$_ACCVIO;
0995      0991      4      END
0996      0992      4      ELSE RETURN SS$_ACCVIO;
0997      0993      4      END;
0998      0994      4
0999      0995      4      IF .LINE_INDENT NEQA 0
1000      0996      4      THEN IF PROBER (%REF (0), %REF (4), .LINE_INDENT)
1001      0997      4      THEN INDENT = .LINE_INDENT
1002      0998      4      ELSE RETURN SS$_ACCVIO;
1003      0999      4
1004      1000      4      IF .INDENT GTRU 0
1005      1001      4      THEN
1006      1002      4      BEGIN
1007      1003      4      IF .WIDTH GTR 0
1008      1004      4      THEN (IF .INDENT GTRU .WIDTH THEN RETURN SS$_BADPARAM)
1009      1005      4      ELSE (IF .INDENT GTRU .ACL_STRING_LEN THEN RETURN SS$_BUFFEROVF);
1010      1006      4      IF .INDENT GTRU MAX_FMT_ACE THEN RETURN SS$_BUFFEROVF;
1011      1007      4      END;
1012      1008      4
1013      1009      4
1014      1010      4      ! Check to see if an access bit name table was supplied. If so, use it
1015      1011      4      ! rather than the default table.
1016      1012      4
1017      1013      4      BIT_NAME_TABLE = 0;
1018      1014      4      IF .BIT_TABLE NEQA 0
1019      1015      4      THEN IF PROBER (%REF (0), %REF (256), .BIT_TABLE)
1020      1016      4      THEN BIT_NAME_TABLE = .BIT_TABLE
1021      1017      4      ELSE RETURN SS$_ACCVIO;
1022      1018      4
1023      1019      4      ! Start building the text ACE.
1024      1020      4
1025      1021      4      CH$FILL (%C ' ', .INDENT, BUFFER);
1026      1022      4      SIZE = LINE_SIZE = .INDENT;
1027      1023      4      STORE TEXT (%C '(');
1028      1024      4      IF PROBER (%REF (0), %REF (DSC$_S_BLN), .ACL_ENTRY)
1029      1025      4      THEN
1030      1026      4      BEGIN
1031      1027      4      ACL_ENTRY_LEN = .ACL_ENTRY[DSC$_W_LENGTH];
1032      1028      4      IF .ACL_ENTRY_LEN GTRU ATR$_ADDACENT THEN RETURN SS$_IVACL;
1033      1029      4      IF EXESPROBER (0, .ACL_ENTRY_LEN, .ACL_ENTRY[DSC$_A_POINTER])
1034      1030      4      THEN CH$MOVE (.ACL_ENTRY_LEN, .ACL_ENTRY[DSC$_A_POINTER], LOCAL_ACE)
1035      1031      4      ELSE RETURN SS$_ACCVIO;
1036      1032      4      END
1037      1033      4      ELSE RETURN SS$_ACCVIO;
1038      1034      4
1039      1035      4      ! Convert the ACE type code.
1040      1036      4
1041      1037      4      CASE .LOCAL_ACE[ACES$_B_TYPE] FROM ACESC_KEYID TO ACESC_DIRDEF OF
1042      1038      4      SET
1043      1039      4      [ACESC_KEYID]:
1044      1040      4      BEGIN
1045      1041      4      ACCESS_MASK = 1;
1046      1042      4      STORE TEXT ('IDENTIFIER=');
1047      1043      4      INCR J FROM .LOCAL_ACE[ACES$_V_RESERVED] + 1 TO (.LOCAL_ACE[ACES$_B_SIZE] - ACESC_LENGTH + 3) / 4
1048      1044      4      DO
1049      1045      4      BEGIN
1050      1046      4      KEY_IDENTIFIER = .VECTOR [LOCAL_ACE[ACES$_L_KEY], .J - 1];
```



```
1051 1047 4      FAO_DESC[DSCSW_LENGTH] = MAX FAO_LENGTH;      ! Max size of an identifier
1052 1048 4      FAO_DESC[DSCSA_POINTER] = FAO_BUF;
1053 1049 4      $FAOL (CTRSTR = $DESCRIPTOR ('!X1'),
1054 1050 4          OUTLEN = FAO_DESC,
1055 1051 4          OUTBUF = FAO_DESC,
1056 1052 4          PRMLST = KEY-IDENTIFIER);
1057 1053 4      CHECK WIDTH (.FAO_DESC[DSCSW_LENGTH]);
1058 1054 4      CHSMOVE (.FAO_DESC[DSCSW_LENGTH],
1059 1055 4          .FAO_DESC[DSCSA_POINTER],
1060 1056 4          BUFFER[.SIZE]);
1061 1057 4      SIZE = .SIZE + .FAO_DESC[DSCSW_LENGTH];
1062 1058 4      STORE_TEXT ('+');
1063 1059 4      END;
1064 1060 4      BUFFER[.SIZE - 1] = %C',';
1065 1061 4      END;
1066 1062 4      [ACESC_BIJNL,
1067 1063 4      ACESC_AIJNL,
1068 1064 4      ACESC_ATJNL]:
1069 1065 4      BEGIN
1070 1066 4      IF .LOCAL ACE[ACESB_TYPE] EQL ACESC_BIJNL
1071 1067 4      THEN STORE_TEXT ('BI JOURNAL=');
1072 1068 4      IF .LOCAL ACE[ACESB_TYPE] EQL ACESC_AIJNL
1073 1069 4      THEN STORE_TEXT ('AI JOURNAL=');
1074 1070 4      IF .LOCAL ACE[ACESB_TYPE] EQL ACESC_ATJNL
1075 1071 4      THEN STORE_TEXT ('AT JOURNAL=');
1076 1072 4      CHECK WIDTH (.LOCAL ACE[ACESB_SIZE] - $BYTEOFFSET (ACESL_ACCESS));
1077 1073 4      CHSMOVE (.LOCAL ACE[ACESB_SIZE] - $BYTEOFFSET (ACESL_ACCESS),
1078 1074 4          LOCAL ACE[ACESL_ACCESS],
1079 1075 4          BUFFER[.SIZE]);
1080 1076 4      SIZE = .SIZE + .LOCAL ACE[ACESB_SIZE] - $BYTEOFFSET (ACESL_ACCESS);
1081 1077 4      STORE_TEXT (',');
1082 1078 4      END;
1083 1079 4      [ACESC_AUDIT,
1084 1080 4      ACESC_ALARM]:
1085 1081 4      BEGIN
1086 1082 4      ACCESS_MASK = 1;
1087 1083 4      AUDIT_MASK = 1;
1088 1084 4      IF .LOCAL ACE[ACESB_TYPE] EQL ACESC_AUDIT
1089 1085 4      THEN STORE_TEXT ('AUDIT JOURNAL=');
1090 1086 4      IF .LOCAL ACE[ACESB_TYPE] EQL ACESC_ALARM
1091 1087 4      THEN STORE_TEXT ('ALARM JOURNAL=');
1092 1088 4      CHECK WIDTH (.LOCAL ACE[ACESB_SIZE] - ACESC_LENGTH);
1093 1089 4      CHSMOVE (.LOCAL ACE[ACESB_SIZE] - ACESC_LENGTH,
1094 1090 4          LOCAL ACE[ACESL_KEY],
1095 1091 4          BUFFER[.SIZE]);
1096 1092 4      SIZE = .SIZE + .LOCAL ACE[ACESB_SIZE] - ACESC_LENGTH;
1097 1093 4      STORE_TEXT (',');
1098 1094 4      END;
1099 1095 4      [ACESC_DIRDEF]:
1100 1096 4      BEGIN
1101 1097 4      STORE_TEXT ('DEFAULT_PROTECTION,');
1102 1098 4      INCR R FROM 0 TO 3
1103 1099 4      DO
1104 1100 4      BEGIN
1105 1101 4      CASE .K FROM 0 TO 3 OF
1106 1102 4      SET
1107 1103 4      [0]: BEGIN
```

```
1108      1104 5      PROT_VALUE = .LOCAL ACE[ACE$! SYS PROT];
1109      1105 5      PROT_FIELD_DSC = $DESCRIPTOR ('SYSTEM:');
1110      1106 4      END;
1111      1107 5      [1]: BEGIN
1112      1108 5      PROT_VALUE = .LOCAL ACE[ACE$! OWN PROT];
1113      1109 5      PROT_FIELD_DSC = $DESCRIPTOR ('OWNER:');
1114      1110 4      END;
1115      1111 5      [2]: BEGIN
1116      1112 5      PROT_VALUE = .LOCAL ACE[ACE$! GRP PROT];
1117      1113 5      PROT_FIELD_DSC = $DESCRIPTOR ('GROUP:');
1118      1114 4      END;
1119      1115 5      [3]: BEGIN
1120      1116 5      PROT_VALUE = .LOCAL ACE[ACE$! WOR PROT];
1121      1117 5      PROT_FIELD_DSC = $DESCRIPTOR ('WORLD:');
1122      1118 4      END;
1123      1119 4      TES:
1124      1120 4      PROT_IDX = 0;
1125      1121 4      INCR J FROM 0 TO 31
1126      1122 4      DO
1127      1123 5      BEGIN
1128      1124 5      IF .PROT_CODE[J] NEQ 0 AND NOT .PROT_VALUE<.J, 1>
1129      1125 5      THEN
1130      1126 6      BEGIN
1131      1127 6      PROT_BUF[.PROT_IDX] = .PROT_CODE[J];
1132      1128 5      PROT_IDX = .PROT_IDX + 1;
1133      1129 5      END;
1134      1130 4      END;
1135      1131 4      CHECK_WIDTH (.PROT_FIELD_DSC[DSC$W_LENGTH] + .PROT_IDX);
1136      1132 4      CH$COPY (.PROT_FIELD_DSC[DSC$W_LENGTH], .PROT_FIELD_DSC[DSC$A_POINTER],
1137      1133 4      .PROT_IDX, PROT_BUF,
1138      1134 4      0,
1139      1135 4      512 - .SIZE,
1140      1136 4      BUFFER[.SIZE]);
1141      1137 4      SIZE = .SIZE + .PROT_FIELD_DSC[DSC$W_LENGTH] + .PROT_IDX;
1142      1138 4      STORE_TEXT (' ');
1143      1139 4      END;
1144      1140 4      END;
1145      1141 4      [ACE$C JNLID]:
1146      1142 4      BEGIN
1147      1143 4      STORE_TEXT ('RMS JOURNAL_ID,');
1148      1144 4      CH$FILL (0, DSC$C S BLN, VOLNAM_DESC);
1149      1145 4      CH$FILL (0, VOLNAM_SIZE, VOLNAM_TEXT);
1150      1146 4      CH$FILL (0, DSC$C S BLN, FILENAME_DESC);
1151      1147 4      CH$FILL (0, ATR$S-FILE SPEC, FILENAME_TEXT);
1152      1148 4      CH$COPY (%CHARCOURT ('DISK$'), UPLIT ('DISK$'),
1153      1149 4      ACCESS VOLNAM, LOCAL ACE[ACE$! VOLNAM],
1154      1150 4      0, VOLNAM_SIZE, VOLNAM_TEXT);
1155      1151 4      VOLNAM_DESC[DSC$W_LENGTH] = CH$FIND CH (VOLNAM_SIZE, VOLNAM_TEXT, 0) -
1156      1152 4      VOLNAM_TEXT;
1157      1153 4      VOLNAM_DESC[DSC$A_POINTER] = VOLNAM_TEXT;
1158      1154 4      FILENAME_DESC[DSC$W_LENGTH] = ATR$S-FILE SPEC;
1159      1155 4      FILENAME_DESC[DSC$A_POINTER] = FILENAME_TEXT;
1160      1156 4      LOCAL_STATUS = LIB$FID_TO_NAME (VOLNAM_DESC, LOCAL ACE[ACE$! FID],
1161      1157 4      FILENAME_DESC, FILENAME_DESC);
1162      1158 4      STORE_TEXT ('JOURNALED_FILE=');
1163      1159 4      IF .LOCAL_STATUS
1164      1160 4      THEN
```

```
1165      1161  4      BEGIN
1166      1162  4      LOCAL      SEGMENT_START : REF VECTOR [,BYTE],
1167      1163  4      SEGMENT_SIZE;      ! Size of segment to get
1168      1164  4      SEGMENT_START = .FILENAME_DESC[DSC$A_POINTER];
1169      1165  4      SEGMENT_SIZE = MINU (.WIDTH - .LINE_SIZE, .FILENAME_DESC[DSC$W_LENGTH]);
1170      1166  4      DO
1171      1167  5          BEGIN
1172      1168  5          IF .SEGMENT_SIZE LSSU .FILENAME_DESC[DSC$W_LENGTH]
1173      1169  5          THEN
1174      1170  5              DECR J FROM .SEGMENT_SIZE TO 1
1175      1171  5              DO
1176      1172  6                  BEGIN
1177      1173  6                  IF .SEGMENT_START[.J - 1] EQL ' '
1178      1174  6                  OR .SEGMENT_START[.J - 1] EQL ']'
1179      1175  6                  OR .SEGMENT_START[.J - 1] EQL ']'
1180      1176  6                  OR .SEGMENT_START[.J - 1] EQL ']'
1181      1177  6                  THEN
1182      1178  7                      BEGIN
1183      1179  7                      SEGMENT_SIZE = .J;
1184      1180  7                      EXITLOOP;
1185      1181  6                      END;
1186      1182  5                  END;
1187      1183  5          CHSMOVE (.SEGMENT_SIZE, .SEGMENT_START, BUFFER[.SIZE]);
1188      1184  5          LINE_SIZE = .LINE_SIZE + .SEGMENT_SIZE;
1189      1185  5          SIZE = .SIZE + .SEGMENT_SIZE;
1190      1186  5          FILENAME_DESC[DSC$W_LENGTH] = .FILENAME_DESC[DSC$W_LENGTH] - .SEGMENT_SIZE;
1191      1187  5          SEGMENT_START = .SEGMENT_START + .SEGMENT_SIZE;
1192      1188  5          IF .FILENAME_DESC[DSC$W_LENGTH] GTR 0 THEN NEW LINE;
1193      1189  5          SEGMENT_SIZE = MINU (.WIDTH - .LINE_SIZE, .FILENAME_DESC[DSC$W_LENGTH]);
1194      1190  5          END
1195      1191  4      UNTIL .FILENAME_DESC[DSC$W_LENGTH] LEQ 0;
1196      1192  4      STORE_TEXT (' ');
1197      1193  4      END
1198      1194  3      ELSE
1199      1195  4          BEGIN
1200      1196  4          FAO_DESC[DSC$W_LENGTH] = MAX FAO_LENGTH;
1201      1197  4          FAO_DESC[DSC$A_POINTER] = FAO_BUF;
1202      1198  4          $FAO ($DESCRIPTOR ('(!UW,!UW,TUW)'),
1203      1199  4              FAO_DESC,
1204      1200  4              FAO_DESC,
1205      1201  4              .((LOCAL_ACE[ACE$T_FID] + $BYTEOFFSET (FID$W_NUM)),
1206      1202  4              .((LOCAL_ACE[ACE$T_FID] + $BYTEOFFSET (FID$W_SEQ)),
1207      1203  4              .((LOCAL_ACE[ACE$T_FID] + $BYTEOFFSET (FID$W_RVN))));
1208      1204  4          CHECK WIDTH (.FAO_DESC[DSC$W_LENGTH]);
1209      1205  4          CHSMOVE (.FAO_DESC[DSC$W_LENGTH], .FAO_DESC[DSC$A_POINTER], BUFFER[.SIZE]);
1210      1206  4          SIZE = .SIZE + .FAO_DESC[DSC$W_LENGTH];
1211      1207  3          END;
1212      1208  3          STORE_TEXT ('MARKED FOR JOURNALING=');
1213      1209  3          FAO_DESC[DSC$W_LENGTH] = MAX FAO_LENGTH;
1214      1210  3          FAO_DESC[DSC$A_POINTER] = FAO_BUF;
1215      1211  3          $FAO ($DESCRIPTOR ('!XD,'),
1216      1212  3              FAO_DESC,
1217      1213  3              FAO_DESC,
1218      1214  3              LOCAL_ACE[ACE$Q_ID_DATE]);
1219      1215  3          FAO_BUF[11] = ' ';
1220      1216  3          IF .FAO_BUF[0] EQL ' '
1221      1217  3          THEN
```

```
1222 1218 4 BEGIN
1223 1219 4 FAO_DESC[DSC$W_LENGTH] = .FAO_DESC[DSC$W_LENGTH] - 1;
1224 1220 4 FAO_DESC[DSC$A_POINTER] = .FAO_DESC[DSC$A_POINTER] + 1;
1225 1221 4 END;
1226 1222 4 CHECK_WIDTH (.FAO_DESC[DSC$W_LENGTH]);
1227 1223 4 CH$MOVE (.FAO_DESC[DSC$W_LENGTH], .FAO_DESC[DSC$A_POINTER], BUFFER[.SIZE]);
1228 1224 4 SIZE = .SIZE + .FAO_DESC[DSC$W_LENGTH];
1229 1225 4 END;
1230 1226 4
1231 1227 4 [INRANGE,
1232 1228 4 OUTRANGE]:
1233 1229 4 BEGIN
1234 1230 4 STORE_TEXT ('Unknown=');
1235 1231 4 CHECK_WIDTH (5);
1236 1232 4 FAO_DESCR[DSC$W_LENGTH] = 5;
1237 1233 4 FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
1238 1234 4 $FAOL (CTRSTR = $DESCRIPTOR ('X!XB,'),
1239 1235 4 OUTBUF = FAO_DESCR,
1240 1236 4 PRMLST = %REF (.LOCAL_ACE[ACESB_TYPE]));
1241 1237 4 SIZE = .SIZE + 5;
1242 1238 4 STORE_TEXT ('Size=');
1243 1239 4 CHECK_WIDTH (5);
1244 1240 4 FAO_DESCR[DSC$W_LENGTH] = 5;
1245 1241 4 FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
1246 1242 4 $FAOL (CTRSTR = $DESCRIPTOR ('D!UB,'),
1247 1243 4 OUTBUF = FAO_DESCR,
1248 1244 4 PRMLST = %REF (.LOCAL_ACE[ACESB_SIZE]));
1249 1245 4 SIZE = .SIZE + 5;
1250 1246 4 STORE_TEXT ('Flags=');
1251 1247 4 CHECK_WIDTH (7);
1252 1248 4 FAO_DESCR[DSC$W_LENGTH] = 7;
1253 1249 4 FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
1254 1250 4 $FAOL (CTRSTR = $DESCRIPTOR ('X!XW,'),
1255 1251 4 OUTBUF = FAO_DESCR,
1256 1252 4 PRMLST = %REF (.LOCAL_ACE[ACESW_FLAGS]));
1257 1253 4 SIZE = .SIZE + 7;
1258 1254 4 STORE_TEXT ('Access=');
1259 1255 4 CHECK_WIDTH (11);
1260 1256 4 FAO_DESCR[DSC$W_LENGTH] = 11;
1261 1257 4 FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
1262 1258 4 $FAOL (CTRSTR = $DESCRIPTOR ('X!XL,'),
1263 1259 4 OUTBUF = FAO_DESCR,
1264 1260 4 PRMLST = %REF (.LOCAL_ACE[ACESL_ACCESS]));
1265 1261 4 SIZE = .SIZE + 11;
1266 1262 4 STORE_TEXT ('Data=');
1267 1263 4 INCR J FROM 1 TO (.LOCAL_ACE[ACESB_SIZE] - ACESC_LENGTH + 3) / 4
1268 1264 4 DO
1269 1265 4 BEGIN
1270 1266 4 CHECK_WIDTH (11);
1271 1267 4 FAO_DESCR[DSC$W_LENGTH] = 11;
1272 1268 4 FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
1273 1269 4 $FAOL (CTRSTR = $DESCRIPTOR ('X!XL,'),
1274 1270 4 OUTBUF = FAO_DESCR,
1275 1271 4 PRMLST = VECTOR [LOCAL_ACE[ACESL_KEY], .J - 1]);
1276 1272 4 SIZE = .SIZE + 11;
1277 1273 4 END;
1278 1274 4 BUFFER[.SIZE - 1] = %C(');
```


\$FORMAT_ACL system service

```
1279 1275 3 IF PROBER (%REF (0), %REF (DSC$S_BLN), .ACL_STRING)
1280 1276 3 THEN
1281 1277 4 BEGIN
1282 1278 4 IF EXESPROBEW (0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
1283 1279 4 THEN CH$COPY (.SIZE, BUFFER, 0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
1284 1280 4 ELSE RETURN SS$ACCVIO
1285 1281 4 END
1286 1282 3 ELSE RETURN SS$ACCVIO;
1287 1283 3 IF .ACL_LENGTH NEQ 0
1288 1284 3 THEN IF PROBEW (%REF (0), %REF (4), .ACL_LENGTH)
1289 1285 3 THEN .ACL_LENGTH = .SIZE ELSE RETURN SS$ACCVIO;
1290 1286 3 IF .SIZE GTR .ACL_STRING_LEN
1291 1287 3 THEN RETURN SS$BUFFEROVF ELSE RETURN SS$NORMAL;
1292 1288 2 END;
1293 1289 2 TES;
1294 1290 2
1295 1291 2 ! Note any special flags applied to the ACE.
1296 1292 2
1297 1293 2 FLAGS = .LOCAL_ACE[ACESW_FLAGS];
1298 1294 2 IF .AUDIT_MASK
1299 1295 2 THEN FLAGS = .FLAGS AND NOT ($FIELDMASK (ACESV_SUCCESS) OR $FIELDMASK (ACESV_FAILURE));
1300 1296 2 IF .FLAGS NEQ 0
1301 1297 2 THEN
1302 1298 3 BEGIN
1303 1299 3 STORE TEXT ('OPTIONS=');
1304 1300 3 IF TESTBITSC (FLAGS[$BITPOSITION (ACESV_DEFAULT)])
1305 1301 3 THEN STORE TEXT ('DEFAULT+');
1306 1302 3 IF TESTBITSC (FLAGS[$BITPOSITION (ACESV_HIDDEN)])
1307 1303 3 THEN STORE TEXT ('HIDDEN+');
1308 1304 3 IF TESTBITSC (FLAGS[$BITPOSITION (ACESV_PROTECTED)])
1309 1305 3 THEN STORE TEXT ('PROTECTED+');
1310 1306 3 IF TESTBITSC (FLAGS[$BITPOSITION (ACESV_NOPROPAGATE)])
1311 1307 3 THEN STORE TEXT ('NOPROPAGATE+');
1312 1308 3 IF .FLAGS NEQ 0
1313 1309 3 THEN
1314 1310 4 BEGIN
1315 1311 4 CHECK_WIDTH (7);
1316 1312 4 FAO_DESCR[DSC$W_LENGTH] = 7;
1317 1313 4 FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
1318 1314 4 $FAO ($DESCRIPTOR ('%X!XW,'),
1319 1315 4 FAO_DESCR,
1320 1316 4 FAO_DESCR,
1321 1317 4 .FLAGS);
1322 1318 4 SIZE = .SIZE + 7;
1323 1319 4 END
1324 1320 3 ELSE BUFFER[.SIZE - 1] = %C',';
1325 1321 3 END;
1326 1322 2
1327 1323 2 ! Note the access rights.
1328 1324 2
1329 1325 2 IF .ACCESS_MASK
1330 1326 2 THEN
1331 1327 3 BEGIN
1332 1328 3 IF .LOCAL_ACE[ACESL_ACCESS] NEQ 0
1333 1329 3 OR (.AUDIT_MASK
1334 1330 3 AND (.LOCAL_ACE[ACESV_SUCCESS]
1335 1331 3 OR .LOCAL_ACE[ACESV_FAILURE]))
```

```
1336 1332 3 THEN
1337 1333 4 BEGIN
1338 1334 4 STORE TEXT ('ACCESS=');
1339 1335 4 INCR J FROM 0 TO 31
1340 1336 4 DO
1341 1337 5 BEGIN
1342 1338 5 IF .(LOCAL_ACE[ACE$$_ACCESS])<.J,1>
1343 1339 5 THEN
1344 1340 6 BEGIN
1345 1341 6 IF .BIT_NAME_TABLE NEQA 0
1346 1342 6 THEN
1347 1343 7 BEGIN
1348 1344 7 IF PROBER (XREF (0), XREF (DSC$_S_BLN), BIT_NAME_TABLE[J, 0, 0, 0, 0])
1349 1345 7 THEN BIT_NAME_DESC = BIT_NAME_TABLE[J, 0, 0, 0, 0]
1350 1346 7 ELSE RETURN SS$_ACCVIO;
1351 1347 7 IF NOT EX$PROBER (0, .BIT_NAME_DESC[DSC$_LENGTH],
1352 1348 7 .BIT_NAME_DESC[DSC$_POINTER])
1353 1349 7 THEN RETURN SS$_ACCVIO;
1354 1350 7 END
1355 1351 6 ELSE BIT_NAME_DESC = .DEFAULT_BITS[J];
1356 1352 6 CHECK_WIDTH (.BIT_NAME_DESC[DSC$_LENGTH] + 1);
1357 1353 6 CH$MOVE (.BIT_NAME_DESC[DSC$_LENGTH], .BIT_NAME_DESC[DSC$_POINTER],
1358 1354 6 .BUFFER[.SIZE]);
1359 1355 6 .BUFFER[.SIZE + .BIT_NAME_DESC[DSC$_LENGTH]] = '+';
1360 1356 6 SIZE = .SIZE + .BIT_NAME_DESC[DSC$_LENGTH] + 1;
1361 1357 5 END;
1362 1358 4 END;
1363 1359 4 IF .AUDIT_MASK
1364 1360 4 THEN
1365 1361 5 BEGIN
1366 1362 5 IF .LOCAL_ACE[ACE$_SUCCESS] THEN STORE TEXT ('SUCCESS+');
1367 1363 5 IF .LOCAL_ACE[ACE$_FAILURE] THEN STORE TEXT ('FAILURE+');
1368 1364 4 END;
1369 1365 4 END
1370 1366 3 ELSE STORE TEXT ('ACCESS=NONE+');
1371 1367 3 END;
1372 1368 2 ! Close off the ACE.
1373 1369 2 BUFFER[.SIZE - 1] = %C')';
1374 1370 2 ! Copy the formatted ACE to the user's buffer and return a size if required.
1375 1371 2 IF PROBER (XREF (0), XREF (DSC$_S_BLN), .ACL_STRING)
1376 1372 2 THEN
1377 1373 3 BEGIN
1378 1374 3 IF EX$PROBEW (0, .ACL_STRING_LEN, .ACL_STRING[DSC$_POINTER])
1379 1375 3 THEN CH$COPY (.SIZE, .BUFFER, 0, .ACL_STRING_LEN, .ACL_STRING[DSC$_POINTER])
1380 1376 3 ELSE RETURN SS$_ACCVIO
1381 1377 3 END
1382 1378 2 ELSE RETURN SS$_ACCVIO;
1383 1379 2 IF .ACL_LENGTH NEQ 0
1384 1380 2 THEN IF PROBEW (XREF (0), XREF (4), .ACL_LENGTH)
1385 1381 2 THEN .ACL_LENGTH = .SIZE ELSE RETURN SS$_ACCVIO;
1386 1382 2 IF .SIZE GTR .ACL_STRING_LEN
1387 1383 2 THEN RETURN SS$_BUFFEROVF ELSE RETURN SS$_NORMAL;
1388 1384 2
```

! End of routine SYS\$FORMAT_ACL

```
.PSECT $SPLITS,NOWRT,NOEXE,2

52 00324 P.ADF: .ASCII \R\
57 00325 .ASCII \W\
45 00326 .ASCII \E\
44 00327 .ASCII \D\
43 00328 .ASCII \C\
00# 00329 .BYTE 0[27]
00 3D 52 45 49 46 49 54 4E 45 44 49 00344 P.ADG: .ASCII \(\<0><0><0>
49 25 21 00348 P.ADH: .ASCII \IDENTIFIER=\<0>
00354 P.ADJ: .ASCII \!XI\
00357 .BLKB 1
00000003 00358 P.ADI: .LONG 3
00000000 0035C .ADDRESS P.ADJ
00 3D 4C 41 4E 52 55 4F 4A 5F 49 42 00360 P.ADK: .ASCII \+\<0><0><0>
00 3D 4C 41 4E 52 55 4F 4A 5F 49 41 00364 P.ADL: .ASCII \BI_JOURNAL=\<0>
00 3D 4C 41 4E 52 55 4F 4A 5F 54 41 00370 P.ADM: .ASCII \AI_JOURNAL=\<0>
0037C P.ADN: .ASCII \AT_JOURNAL=\<0>
00 00 00 2C 00388 P.ADO: .ASCII \,\<0><0><0>
0038C P.ADP: .ASCII \AUDIT_JOURNAL=\<0><0>
0039B
00 3D 4C 41 4E 52 55 4F 4A 5F 4D 52 41 4C 41 0039C P.ADQ: .ASCII \ALARM_JOURNAL=\<0><0>
003AB
00 00 00 2C 003AC P.ADR: .ASCII \,\<0><0><0>
54 43 45 54 4F 52 50 5F 54 4C 55 41 46 45 44 00380 P.ADS: .ASCII \DEFAULT_PROTECTION,\<0>
00 2C 4E 4F 49 0038F
3A 4D 45 54 53 59 53 003C4 P.ADU: .ASCII \SYSTEM:\
003CB .BLKB 1
00000007 003CC P.ADT: .LONG 7
00000000 003D0 .ADDRESS P.ADU
3A 52 45 4E 57 4F 003D4 P.ADW: .ASCII \OWNER:\
003DA .BLKB 2
00000006 003DC P.ADV: .LONG 6
00000000 003E0 .ADDRESS P.ADW
3A 50 55 4F 52 47 003E4 P.ADY: .ASCII \GROUP:\
003EA .BLKB 2
00000006 003EC P.ADX: .LONG 6
00000000 003F0 .ADDRESS P.ADY
3A 44 4C 52 4F 57 003F4 P.AEA: .ASCII \WORLD:\
003FA .BLKB 2
00000006 003FC P.ADZ: .LONG 6
00000000 00400 .ADDRESS P.AEA
2C 44 49 5F 4C 41 4E 52 55 4F 4A 5F 53 4D 52 00404 P.AEB: .ASCII \,\<0><0><0>
00 00408 P.AEC: .ASCII \RMS_JOURNAL_ID,\<0>
00417
3D 45 4C 49 46 5F 44 45 4C 41 4E 52 55 4F 4A 00418 P.AED: .ASCII \DISK$\<0><0><0>
00420 P.AEE: .ASCII \JOURNALED_FILE=\<0>
0042F
00 00 00 2C 00430 P.AEF: .ASCII \,\<0><0><0>
2C 29 57 55 21 2C 57 55 21 2C 57 55 21 28 00434 P.AEH: .ASCII \(!UW,!UW,!UW),\
00442 .BLKB 2
0000000E 00444 P.AEG: .LONG 14
00000000 00448 .ADDRESS P.AEH
```


| | | | | | | | | | | | |
|----|----|----|----|-----------|------|-------|-------|--------|--------------------------------------|-------------------------------|------|
| 6E | 56 | 0C | 4F | 00000000G | 6E | D0 | 0001A | MOVL | ACL_STRING_LEN, R1 | : | |
| | | | AC | | 53 | D4 | 0001D | CLRL | R3 | : | |
| | | | 6E | | 00 | 16 | 0001F | JSB | EXESPROBEW | : | |
| | | | | | 50 | E9 | 00025 | BLBC | R0, 2\$ | : | |
| | | | | | 04 | C1 | 00028 | ADDL3 | #4, ACL_STRING, R6 | : | 0963 |
| | | | | | 00 | 2C | 0002D | MOVCS | #0, (SPT, #32, ACL_STRING_LEN, @R6)+ | : | |
| | | | | | 96 | | 00032 | | | : | |
| | | | | | 59 | 7C | 00033 | CLRQ | WIDTH | : | 0970 |
| | | | | 08 | AE | D4 | 00035 | CLRL | TERM_POINTER | : | 0971 |
| | | | | | 58 | D4 | 00038 | CLRL | TERM_LENGTH | : | |
| | | | | 24 | AE | 94 | 0003A | CLRB | AUDIT_MASK | : | 0972 |
| | | | | 20 | AE | 94 | 0003D | CLRB | ACCESS_MASK | : | |
| | | | | 10 | AC | D0 | 00040 | MOVL | LINE_WIDTH, R0 | : | 0976 |
| | | | | | 09 | 13 | 00044 | BEQL | 1\$ | : | |
| | 60 | | | | 00 | 0C | 00046 | PROBER | #0, #4, (R0) | : | 0977 |
| | | | | | 74 | 13 | 0004A | BEQL | 11\$ | : | |
| | | | | | 60 | D0 | 0004C | MOVL | (R0), WIDTH | : | 0978 |
| | | | | | AC | D5 | 0004F | TSTL | TERM_DESC | : | 0981 |
| | | | | 14 | 29 | 13 | 00052 | BEQL | 4\$ | : | |
| | | | | | 00 | 0C | 00054 | PROBER | #0, #8, @TERM_DESC | : | 0984 |
| | | | | | 65 | 13 | 00059 | BEQL | 11\$ | : | |
| | | | | | BC | 3C | 0005B | MOVZWL | @TERM_DESC, TERM_LENGTH | : | 0987 |
| | | | | | 04 | C1 | 0005F | ADDL3 | #4, TERM_DESC, R0 | : | 0988 |
| | | | | | 60 | D0 | 00064 | MOVL | (R0), TERM_POINTER | : | |
| | | | | | AE | D0 | 00068 | MOVL | TERM_POINTER, R0 | : | 0989 |
| | | | | | 50 | | 0006C | MOVL | TERM_LENGTH, R1 | : | |
| | | | | | 51 | 5B | 0006C | MOVL | TERM_LENGTH, R1 | : | |
| | | | | | 53 | D4 | 0006F | CLRL | R3 | : | |
| | | | | | 00 | 16 | 00071 | JSB | EXESPROBER | : | |
| | | | | | 50 | E8 | 00077 | BLBS | R0, 4\$ | : | |
| | | | | | 31 | 0007A | 3\$: | BRW | 172\$ | : | |
| | | | | | AC | D0 | 0007D | 4\$: | MOVL | LINE_INDENT, R0 | 0995 |
| | | | | | 09 | 13 | 00081 | BEQL | 6\$ | : | |
| | | | | | 00 | 0C | 00083 | PROBER | #0, #4, (R0) | : | 0996 |
| | | | | | F1 | 13 | 00087 | BEQL | 3\$ | : | |
| | | | | | 60 | D0 | 00089 | MOVL | (R0), INDENT | : | 0997 |
| | | | | | 5A | D5 | 0008C | 6\$: | TSTL | INDENT | 1000 |
| | | | | | 1E | 13 | 0008E | BEQL | 10\$ | : | |
| | | | | | 59 | D5 | 00090 | TSTL | WIDTH | : | 1003 |
| | | | | | 09 | 15 | 00092 | BLEQ | 7\$ | : | |
| | | | | | 5A | D1 | 00094 | CMP | INDENT, WIDTH | : | 1004 |
| | | | | | 09 | 1B | 00097 | BLEQU | 8\$ | : | |
| | | | | | 50 | 14 | 00099 | MOVL | #20, R0 | : | |
| | | | | | 04 | 0009C | | RET | | : | |
| | | | | | 5A | D1 | 0009D | 7\$: | CMP | INDENT, ACL_STRING_LEN | 1005 |
| | | | | | 07 | 1A | 000A0 | BGTRU | 9\$ | : | |
| | | | | | 5A | D1 | 000A2 | 8\$: | CMP | INDENT, #3072 | 1006 |
| | | | | | 03 | 1B | 000A9 | 9\$: | BLEQU | 10\$ | |
| | | | | | 105F | 31 | 000AB | BRW | 174\$ | : | |
| | | | | | EF | D4 | 000AE | 10\$: | CLRL | BIT_NAME_TABLE | 1013 |
| | | | | | AC | D0 | 000B4 | MOVL | BIT_TABLE, R0 | : | 1014 |
| | | | | | 0F | 13 | 000B8 | BEQL | 12\$ | : | |
| | | | | | 00 | 0C | 000BA | PROBER | #0, #256, (R0) | : | 1015 |
| | | | | | B8 | 13 | 000C0 | BEQL | 3\$ | : | |
| | | | | | 50 | D0 | 000C2 | MOVL | R0, BIT_NAME_TABLE | : | 1016 |
| | | | | | 00 | 2C | 000C9 | 12\$: | MOVCS | #0, (SPT, #32, INDENT, BUFFER | 1021 |
| | | | | | CE | | 000CE | | | : | |
| | | | | | 5A | D0 | 000D1 | MOVL | INDENT, LINE_SIZE | : | 1022 |

| | | | | | | | | | | | |
|------|------|----------|----------|------|------|------|-------|--------|--|--|------|
| | | 56 | | 28 | 5A | D0 | 000D4 | MOVL | INDENT, SIZE | | |
| | | | | | AE | D4 | 000D7 | CLRL | 40(SP) | | 1023 |
| | | | | | 59 | D5 | 000DA | TSTL | WIDTH | | |
| | | | | | 2E | 13 | 000DC | BEQL | 14\$ | | |
| | | | | 28 | AE | D6 | 000DE | INCL | 40(SP) | | |
| | | 50 | | 01 | A7 | 9E | 000E1 | MOVAB | 1(R7), R0 | | |
| | | 59 | | | 50 | D1 | 000E5 | CMPL | R0, WIDTH | | |
| | | | | | 22 | 1B | 000E8 | BLEQU | 14\$ | | |
| | | | | | 5B | D5 | 000EA | TSTL | TERM_LENGTH | | |
| | | | | | 0F | 15 | 000EC | BLEQ | 13\$ | | |
| | 0A5C | CE46 | 0B | | 5B | 28 | 000EE | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | | |
| | | | | 14 | BC | 3C | 000F6 | MOVZWL | @TERM_DESC, R0 | | |
| 5A | | | | | 50 | C0 | 000FA | ADDL2 | R0, SIZE | | |
| | 20 | | | | 6E | 00 | 000FD | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | | |
| | | | | | 0A5C | CE46 | 00102 | | | | |
| | | 56 | | | 5A | C0 | 00106 | ADDL2 | INDENT, SIZE | | |
| | | 57 | | | 5A | D0 | 00109 | MOVL | INDENT, LINE_SIZE | | |
| | | | | | 57 | D6 | 0010C | INCL | LINE_SIZE | | |
| | 0A5C | CE46 | 00000000 | | EF | 90 | 0010E | MOVB | P.ADG, BUFFER[SIZE] | | |
| | | | | | 56 | D6 | 00118 | INCL | SIZE | | |
| | | 54 | | 04 | AC | D0 | 0011A | MOVL | ACL_ENTRY, R4 | | 1024 |
| | 64 | 0B | | | 00 | 0C | 0011E | PROBER | #0, #8, (R4) | | |
| | | | | | 24 | 13 | 00122 | BEQL | 16\$ | | |
| | | 55 | | | 64 | 3C | 00124 | MOVZWL | (R4), ACL_ENTRY_LEN | | 1027 |
| | | 000000FF | | | 55 | D1 | 00127 | CMPL | ACL_ENTRY_LEN, #255 | | 1028 |
| | | | | | 06 | 1B | 0012E | BLEQU | 15\$ | | |
| | | 50 | | 21E4 | 8F | 3C | 00130 | MOVZWL | #8676, R0 | | |
| | | | | | 04 | 00 | 00135 | RET | | | |
| | | 50 | | 04 | A4 | D0 | 00136 | MOVL | 4(R4), R0 | | 1029 |
| | | 51 | | | 55 | D0 | 0013A | MOVL | ACL_ENTRY_LEN, R1 | | |
| | | | | | 53 | D4 | 0013D | CLRL | R3 | | |
| | | | | | 00 | 16 | 0013F | JSB | EXESPROBER | | |
| | | 03 | | | 50 | E8 | 00145 | BLBS | R0, 17\$ | | |
| | | | | | 0FB9 | 31 | 00148 | BRW | 172\$ | | |
| | FF00 | CD | 04 | | 55 | 28 | 0014B | MOV3 | ACL_ENTRY_LEN, @4(R4), LOCAL_ACE | | 1030 |
| | | | 18 | | CD | 9A | 00152 | MOVZBL | LOCAL_ACE+1, 24(SP) | | 1037 |
| | | 0B | | | AE | 8F | 00158 | CASEB | 24(SPT), #1, #8 | | |
| 04A8 | | | | | 0387 | | 0015D | .WORD | 46\$-18\$,- | | |
| 08A4 | 04A8 | | 04A8 | | 0609 | | 00165 | | 55\$-18\$,- | | |
| | 0012 | | 0609 | | 072E | | 0016D | | 55\$-18\$,- | | |
| | | | | | | | | | 55\$-18\$,- | | |
| | | | | | | | | | 55\$-18\$,- | | |
| | | | | | | | | | 69\$-18\$,- | | |
| | | | | | | | | | 69\$-18\$,- | | |
| | | | | | | | | | 19\$-18\$,- | | |
| | | | | | | | | | 98\$-18\$,- | | |
| | | | | | | | | | 81\$-18\$,- | | |
| | | 2B | | 2B | AE | E9 | 0016F | BLBC | 40(SP), 21\$ | | 1230 |
| | | 50 | | 0B | A7 | 9E | 00173 | MOVAB | 8(R7), R0 | | |
| | | 59 | | | 50 | D1 | 00177 | CMPL | R0, WIDTH | | |
| | | | | | 22 | 1B | 0017A | BLEQU | 21\$ | | |
| | | | | | 5B | D5 | 0017C | TSTL | TERM_LENGTH | | |
| | | | | | 0F | 15 | 0017E | BLEQ | 20\$ | | |
| | 0A5C | CE46 | 0B | | 5B | 28 | 00180 | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | | |
| | | | | 14 | BC | 3C | 00188 | MOVZWL | @TERM_DESC, R0 | | |
| | | | | | 50 | C0 | 0018C | ADDL2 | R0, SIZE | | |
| 5A | | | | | 6E | 00 | 0018F | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | | |
| | 20 | | | | 0A5C | CE46 | 00194 | | | | |

| | | | | | | | | | |
|----|----|---------------------|----|-----------|----|-------|-------------|--|------|
| 5A | 20 | 0A5C CE46 00000000' | 56 | 5A | C0 | 00198 | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 | 0019B | MOVL | INDENT, LINE_SIZE | |
| | | | 57 | 08 | C0 | 0019E | 21\$: ADDL2 | #8, LINE_SIZE | |
| | | | EF | 08 | 28 | 001A1 | MOVCL | #8, P.AEL, BUFFER[SIZE] | |
| | | | 56 | 08 | C0 | 001AC | ADDL2 | #8, SIZE | |
| | | | 2B | AE | E9 | 001AF | BLBC | 40(SP), 23\$ | 1231 |
| | | | 50 | A7 | 9E | 001B3 | MOVAB | 5(R7), R0 | |
| | | | 59 | 50 | D1 | 001B7 | CMPL | R0, WIDTH | |
| | | | | 22 | 1B | 001BA | BLEQU | 23\$ | |
| | | | | 5B | D5 | 001BC | TSTL | TERM_LENGTH | |
| | | | | 0F | 15 | 001BE | BLEQ | 22\$ | |
| | | 0A5C CE46 08 | BE | 5B | 28 | 001C0 | MOVCL | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 50 | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | | | 56 | 50 | C0 | 001CC | ADDL2 | R0, SIZE | |
| | | | 6E | 00 | 2C | 001CF | 22\$: MOVCL | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | 0A5C CE46 | | 001D4 | | | |
| | | | 56 | 5A | C0 | 001D8 | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 | 001DB | MOVL | INDENT, LINE_SIZE | |
| | | | 57 | 05 | C0 | 001DE | 23\$: ADDL2 | #5, LINE_SIZE | |
| | | FEF8 | CD | 05 | B0 | 001E1 | MOVW | #5, FAO_DESCR | 1232 |
| | | FEFC | CD | 0A5C CE46 | 9E | 001E6 | MOVAB | BUFFER[SIZE], FAO_DESCR+4 | 1233 |
| | | 1C | AE | 18 | AE | D0 | MOVL | 24(SP), 28(SP) | 1236 |
| | | | | 1C | AE | 9F | PUSHAB | 28(SP) | |
| | | | | FEF8 | CD | 9F | PUSHAB | FAO_DESCR | |
| | | | | | 7E | D4 | CLRL | -(SP) | |
| | | | | 00000000' | EF | 9F | PUSHAB | P.AEM | |
| | | | 00 | 04 | FB | 00202 | CALLS | #4, SYSSFAOL | |
| | | | 56 | 05 | C0 | 00209 | ADDL2 | #5, SIZE | 1237 |
| | | | 2B | 28 | AE | E9 | BLBC | 40(SP), 25\$ | 1238 |
| | | | 50 | 05 | A7 | 9E | MOVAB | 5(R7), R0 | |
| | | | 59 | 50 | D1 | 00214 | CMPL | R0, WIDTH | |
| | | | | 22 | 1B | 00217 | BLEQU | 25\$ | |
| | | | | 5B | D5 | 00219 | TSTL | TERM_LENGTH | |
| | | | | 0F | 15 | 0021B | BLEQ | 24\$ | |
| | | 0A5C CE46 08 | BE | 5B | 28 | 0021D | MOVCL | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 50 | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | | | 56 | 50 | C0 | 00225 | ADDL2 | R0, SIZE | |
| | | | 6E | 00 | 2C | 0022C | 24\$: MOVCL | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | 0A5C CE46 | | 00231 | | | |
| | | | 56 | 5A | C0 | 00235 | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 | 00238 | MOVL | INDENT, LINE_SIZE | |
| | | | 57 | 05 | C0 | 0023B | 25\$: ADDL2 | #5, LINE_SIZE | |
| | | 0A5C CE46 00000000' | EF | 05 | 28 | 0023E | MOVCL | #5, P.AEO, BUFFER[SIZE] | |
| | | | 56 | 05 | C0 | 00249 | ADDL2 | #5, SIZE | |
| | | | 2B | 28 | AE | E9 | BLBC | 40(SP), 27\$ | 1239 |
| | | | 50 | 05 | A7 | 9E | MOVAB | 5(R7), R0 | |
| | | | 59 | 50 | D1 | 00254 | CMPL | R0, WIDTH | |
| | | | | 22 | 1B | 00257 | BLEQU | 27\$ | |
| | | | | 5B | D5 | 00259 | TSTL | TERM_LENGTH | |
| | | | | 0F | 15 | 0025B | BLEQ | 26\$ | |
| | | 0A5C CE46 08 | BE | 5B | 28 | 0025D | MOVCL | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 50 | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | | | 56 | 50 | C0 | 00269 | ADDL2 | R0, SIZE | |
| | | | 6E | 00 | 2C | 0026C | 26\$: MOVCL | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | 0A5C CE46 | | 00271 | | | |
| | | | 56 | 5A | C0 | 00275 | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 | 00278 | MOVL | INDENT, LINE_SIZE | |

| | | | | | | | | | | | |
|----|-----------|------|-----------|------|------|------|-------|-------|--------|--|------|
| | | | 57 | | 05 | C0 | 0027B | 27\$: | ADDL2 | #5, LINE SIZE | |
| | FEF8 | CD | | | 05 | B0 | 0027E | | MOVW | #5, FAO_DESCR | 1240 |
| | FEFC | CD | | OASC | CE46 | 9E | 00283 | | MOVAB | BUFFER[SIZE], FAO_DESCR+4 | 1241 |
| | 1C | AE | | FF00 | CD | 9A | 0028B | | MOVZBL | LOCAL_ACE, 28(SP) | 1244 |
| | | | | 1C | AE | 9F | 00291 | | PUSHAB | 28(SP) | |
| | | | | FEF8 | CD | 9F | 00294 | | PUSHAB | FAO_DESCR | |
| | | | | | 7E | D4 | 00298 | | CLRL | -(SP) | |
| | | | | | EF | 9F | 0029A | | PUSHAB | P.AEP | |
| | 00000000G | 00 | | | 04 | FB | 002A0 | | CALLS | #4, SYSSFAOL | |
| | | 56 | | | 05 | C0 | 002A7 | | ADDL2 | #5, SIZE | 1245 |
| | | 28 | | | AE | E9 | 002AA | | BLBC | 40(SP), 29\$ | 1246 |
| | | 50 | | | A7 | 9E | 002AE | | MOVAB | 6(R7), R0 | |
| | | 59 | | | 50 | D1 | 002B2 | | CMPL | R0, WIDTH | |
| | | | | | 22 | 1B | 002B5 | | BLEQU | 29\$ | |
| | | | | | 5B | D5 | 002B7 | | TSTL | TERM_LENGTH | |
| | | | | | 0F | 15 | 002B9 | | BLEQ | 28\$ | |
| | OASC | CE46 | 08 | BE | 5B | 28 | 002BB | | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | | 50 | BC | 3C | 002C3 | | MOVZWL | @TERM_DESC, R0 | |
| | | | | 56 | 50 | C0 | 002C7 | | ADDL2 | R0, SIZE | |
| SA | | 20 | 6E | | 00 | 2C | 002CA | 28\$: | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | | OASC | CE46 | 002CF | | | | |
| | | | | 56 | 5A | C0 | 002D3 | | ADDL2 | INDENT, SIZE | |
| | | | | 57 | 5A | D0 | 002D6 | | MOVL | INDENT, LINE SIZE | |
| | | | | 57 | 06 | C0 | 002D9 | 29\$: | ADDL2 | #6, LINE SIZE | |
| | OASC | CE46 | 00000000' | EF | 06 | 28 | 002DC | | MOV3 | #6, P.AEP, BUFFER[SIZE] | |
| | | | | 56 | 06 | C0 | 002E7 | | ADDL2 | #6, SIZE | |
| | | | | 2B | AE | E9 | 002EA | | BLBC | 40(SP), 31\$ | 1247 |
| | | | | 50 | A7 | 9E | 002EE | | MOVAB | 7(R7), R0 | |
| | | | | 59 | 50 | D1 | 002F2 | | CMPL | R0, WIDTH | |
| | | | | | 22 | 1B | 002F5 | | BLEQU | 31\$ | |
| | | | | | 5B | D5 | 002F7 | | TSTL | TERM_LENGTH | |
| | | | | | 0F | 15 | 002F9 | | BLEQ | 30\$ | |
| | OASC | CE46 | 08 | BE | 5B | 28 | 002FB | | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | | 50 | BC | 3C | 00303 | | MOVZWL | @TERM_DESC, R0 | |
| | | | | 56 | 50 | C0 | 00307 | | ADDL2 | R0, SIZE | |
| SA | | 20 | 6E | | 00 | 2C | 0030A | 30\$: | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | | OASC | CE46 | 0030F | | | | |
| | | | | 56 | 5A | C0 | 00313 | | ADDL2 | INDENT, SIZE | |
| | | | | 57 | 5A | D0 | 00316 | | MOVL | INDENT, LINE SIZE | |
| | | | | 57 | 07 | C0 | 00319 | 31\$: | ADDL2 | #7, LINE SIZE | |
| | FEF8 | CD | | | 07 | B0 | 0031C | | MOVW | #7, FAO_DESCR | 1248 |
| | FEFC | CD | | OASC | CE46 | 9E | 00321 | | MOVAB | BUFFER[SIZE], FAO_DESCR+4 | 1249 |
| | 1C | AE | | FF02 | CD | 3C | 00329 | | MOVZWL | LOCAL_ACE+2, 28(SP) | 1252 |
| | | | | 1C | AE | 9F | 0032F | | PUSHAB | 28(SP) | |
| | | | | FEF8 | CD | 9F | 00332 | | PUSHAB | FAO_DESCR | |
| | | | | | 7E | D4 | 00336 | | CLRL | -(SP) | |
| | | | | | EF | 9F | 00338 | | PUSHAB | P.AES | |
| | 00000000G | 00 | | | 04 | FB | 0033E | | CALLS | #4, SYSSFAOL | |
| | | 56 | | | 07 | C0 | 00345 | | ADDL2 | #7, SIZE | 1253 |
| | | 28 | | | AE | E9 | 00348 | | BLBC | 40(SP), 33\$ | 1254 |
| | | 50 | | | A7 | 9E | 0034C | | MOVAB | 7(R7), R0 | |
| | | 59 | | | 50 | D1 | 00350 | | CMPL | R0, WIDTH | |
| | | | | | 22 | 1B | 00353 | | BLEQU | 33\$ | |
| | | | | | 5B | D5 | 00355 | | TSTL | TERM_LENGTH | |
| | | | | | 0F | 15 | 00357 | | BLEQ | 32\$ | |
| | OASC | CE46 | 08 | BE | 5B | 28 | 00359 | | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | | 50 | BC | 3C | 00361 | | MOVZWL | @TERM_DESC, R0 | |

| | | | | | | | | | |
|----|---------------------|----------|------------|------------|----------------|-------|----------------|---|------|
| SA | 20 | 56 6E | 50 00 | C0 2C | 00365 00368 | 32\$: | ADDL2 MOVCS | RO, SIZE #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 56 57 | 0A5C 5A | CE46 D0 | 0036D 00371 | | ADDL2 MOVL | INDENT, SIZE INDENT, LINE SIZE | |
| | | 57 | 07 | C0 | 00374 | 33\$: | ADDL2 | #7, LINE SIZE | |
| | 0A5C CE46 00000000' | EF | 07 | 28 | 00377 | | MOVCS | #7, P.AED, BUFFER[SIZE] | |
| | | 56 | 07 | C0 | 00385 | | ADDL2 | #7, SIZE | |
| | | 2B | AE | E9 | 00388 | | BLBC | 40(SP), 35\$ | 1255 |
| | | 50 | A7 | 9E | 0038C | | MOVAB | 11(R7), RO | |
| | | 59 | 50 | D1 | 00390 | | CMPL | RO, WIDTH | |
| | | | 22 | 1B | 00393 | | BLEQU | 35\$ | |
| | | | 5B | D5 | 00395 | | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | 00397 | | BLEQ | 34\$ | |
| | 0A5C CE46 08 | BE | 5B | 28 | 00399 | | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 50 | 14 | BC | 3C | 003A1 | MOVZWL | @TERM_DESC, RO | |
| | | 56 | 50 | C0 | 003A5 | | ADDL2 | RO, SIZE | |
| SA | 20 | 6E | 00 | 2C | 003A8 | 34\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 56 | 0A5C | CE46 | 003AD | | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A | C0 | 003B1 | | MOVL | INDENT, LINE SIZE | |
| | | 57 | 0B | C0 | 003B4 | 35\$: | ADDL2 | #11, LINE SIZE | |
| | | CD | 0B | B0 | 003BA | | MOVW | #11, FAO_DESCR | 1256 |
| | FEF8 | CD | 0A5C | CE46 | 9E | 003BF | MOVAB | BUFFER[SIZE], FAO_DESCR+4 | 1257 |
| | FEFC | CD | FF04 | CD | D0 | 003C7 | MOVL | LOCAL_ACE+4, 28(SP) | 1260 |
| | 1C | AE | 1C | AE | 9F | 003CD | PUSHAB | 28(SP) | |
| | | | FEF8 | CD | 9F | 003D0 | PUSHAB | FAO_DESCR | |
| | | | | 7E | D4 | 003D4 | CLRL | -(SP) | |
| | | | | EF | 9F | 003D6 | PUSHAB | P.AEV | |
| | 00000000G | 00 | 04 | FB | 003DC | | CALLS | #4, SYSSFAOL | |
| | | 56 | 0B | C0 | 003E3 | | ADDL2 | #11, SIZE | 1261 |
| | | 2B | 28 | AE | E9 | 003E6 | BLBC | 40(SP), 37\$ | 1262 |
| | | 50 | 05 | A7 | 9E | 003EA | MOVAB | 5(R7), RO | |
| | | 59 | 50 | D1 | 003EE | | CMPL | RO, WIDTH | |
| | | | 22 | 1B | 003F1 | | BLEQU | 37\$ | |
| | | | 5B | D5 | 003F3 | | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | 003F5 | | BLEQ | 36\$ | |
| | 0A5C CE46 08 | BE | 5B | 28 | 003F7 | | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 50 | 14 | BC | 3C | 003FF | MOVZWL | @TERM_DESC, RO | |
| | | 56 | 50 | C0 | 00403 | | ADDL2 | RO, SIZE | |
| SA | 20 | 6E | 00 | 2C | 00406 | 36\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 56 | 0A5C | CE46 | 0040B | | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A | C0 | 0040F | | MOVL | INDENT, LINE SIZE | |
| | | 57 | 05 | C0 | 00412 | 37\$: | ADDL2 | #5, LINE SIZE | |
| | 0A5C CE46 00000000' | EF | 05 | 28 | 00415 | | MOVCS | #5, P.AEX, BUFFER[SIZE] | |
| | | 56 | 05 | C0 | 00418 | | ADDL2 | #5, SIZE | |
| | | 1C | AE | 9A | 00423 | | MOVZBL | LOCAL_ACE, 28(SP) | 1263 |
| | | 1C | AE | C2 | 0042C | | SUBL2 | #5, 28(SP) | |
| | | 1C | AE | C6 | 00430 | | DIVL2 | #4, 28(SP) | |
| | | | 5B | D4 | 00434 | | CLRL | J | |
| | | | 5A | 11 | 00436 | | BRB | 41\$ | |
| | | 2B | 28 | AE | E9 | 00438 | BLBC | 40(SP), 40\$ | 1266 |
| | | 50 | 0B | A7 | 9E | 0043C | MOVAB | 11(R7), RO | |
| | | 59 | 50 | D1 | 00440 | | CMPL | RO, WIDTH | |
| | | | 22 | 1B | 00443 | | BLEQU | 40\$ | |
| | | | 5B | D5 | 00445 | | TSTL | TERM_LENGTH | |

| | | | | | | | | | |
|----|---------------------|------|-----------|-----------|----|-------|--------|--|-------|
| SA | 0A5C CE46 | 08 | BE | 0F | 15 | 00447 | BLEQ | 39\$ | |
| | | | 50 | 5B | 28 | 00449 | MOV C3 | TERM LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 56 | 14 | 3C | 00451 | MOVZWL | @TERM_DESC, R0 | |
| | | | 6E | 50 | C0 | 00455 | ADDL2 | R0, SIZE | |
| | 20 | | | 00 | 2C | 00458 | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | 56 | 0A5C CE46 | | 0045D | | | |
| | | | 57 | 5A | C0 | 00461 | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 | 00464 | MOVL | INDENT, LINE SIZE | |
| | | | CD | 0B | C0 | 00467 | ADDL2 | #11, LINE SIZE | |
| | FEF8 | | CD | 0B | B0 | 0046A | MOVW | #11, FAO_DESCR | 1267 |
| | FEFC | | CD | 0A5C CE46 | 9E | 0046F | MOVAB | BUFFER[SIZE], FAO_DESCR+4 | 1268 |
| | | | | FF04 CD48 | DF | 00477 | PUSHAL | LOCAL ACE+4[J] | 1271 |
| | | | | FEF8 | CD | 9F | PUSHAB | FAO_DESCR | |
| | | | | 7E | D4 | 00480 | CLRL | -(SP) | |
| | | | | EF | 9F | 00482 | PUSHAB | P.AEY | |
| | 00000000G | 00 | | 04 | FB | 00488 | CALLS | #4, SYSSFAOL | |
| | | 56 | | 0F | C0 | 0048F | ADDL2 | #11, SIZE | 1272 |
| | A1 | 58 | | 1C | AE | F3 | AOBLEQ | 28(SP), J, 38\$ | 1263 |
| | | 58 | | 29 | 90 | 00497 | MOVB | #41, BUFFER-1[SIZE] | 1274 |
| | OC | BC | 0A5B CE46 | 00 | 0C | 0049D | PROBER | #0, #8, @ACL_STRING | 1275 |
| | | | 08 | 16 | 13 | 004A2 | BEQL | 42\$ | |
| | 54 | OC | AC | 04 | C1 | 004A4 | ADDL3 | #4, ACL_STRING, R4 | 1278 |
| | | | 50 | 64 | D0 | 004A9 | MOVL | (R4), R0 | |
| | | | 51 | 6E | D0 | 004AC | MOVL | ACL_STRING_LEN, R1 | |
| | | | | 53 | D4 | 004AF | CLRL | R3 | |
| | | | | 00000000G | 00 | 16 | JSB | EXESPROBEW | |
| | | | D3 | 50 | E8 | 004B7 | BLBS | R0, 43\$ | |
| | | | | OC47 | 31 | 004BA | BRW | 172\$ | |
| | 58 | OC | AC | 04 | C1 | 004BD | ADDL3 | #4, ACL_STRING, R8 | 1279 |
| 6E | 00 | 0A5C | CE | 56 | 2C | 004C2 | MOV C5 | SIZE, BUFFER, #0, ACL_STRING_LEN, @(R8)+ | |
| | | | | 98 | | 004C9 | | | |
| | | | 50 | 08 | AC | D0 | MOVL | ACL_LENGTH, R0 | 1283 |
| | | | | 09 | 13 | 004CE | BEQL | 44\$ | |
| | 60 | | 04 | 00 | 0D | 004D0 | PROBEW | #0, #4, (R0) | 1284 |
| | | | | E4 | 13 | 004D4 | BEQL | 42\$ | |
| | | | 60 | 56 | D0 | 004D6 | MOVL | SIZE, (R0) | 1285 |
| | | | 6E | 56 | D1 | 004D9 | CMPL | SIZE, ACL_STRING_LEN | 1286 |
| | | | | 03 | 14 | 004DC | BGTR | 45\$ | |
| | | | | OC32 | 31 | 004DE | BRW | 175\$ | |
| | | | | OC29 | 31 | 004E1 | BRW | 174\$ | |
| | | | 20 | 01 | 90 | 004E4 | MOVB | #1, ACCESS_MASK | 1041 |
| | | | 2B | AE | E9 | 004E8 | BLBC | 40(SP), 48\$ | 1042 |
| | | | 50 | A7 | 9E | 004EC | MOVAB | 11(R7), R0 | |
| | | | 59 | 50 | D1 | 004F0 | CMPL | R0, WIDTH | |
| | | | | 22 | 1B | 004F3 | BLEQU | 48\$ | |
| | | | | 5B | D5 | 004F5 | TSTL | TERM_LENGTH | |
| | | | | 0F | 15 | 004F7 | BLEQ | 47\$ | |
| | 0A5C CE46 | 08 | BE | 5B | 28 | 004F9 | MOV C3 | TERM LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 50 | 14 | 3C | 00501 | MOVZWL | @TERM_DESC, R0 | |
| | | | 56 | 50 | C0 | 00505 | ADDL2 | R0, SIZE | |
| SA | 20 | | 6E | 00 | 2C | 00508 | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | 47\$: |
| | | | | 0A5C CE46 | | 0050D | | | |
| | | | 56 | 5A | C0 | 00511 | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 | 00514 | MOVL | INDENT, LINE SIZE | |
| | | | 57 | 0B | C0 | 00517 | ADDL2 | #11, LINE SIZE | 48\$: |
| | 0A5C CE46 00000000' | | EF | 0B | 28 | 0051A | MOV C3 | #11, P.ADR, BUFFER[SIZE] | |
| | | | 56 | 0B | C0 | 00525 | ADDL2 | #11, SIZE | |

| | | | | | | | | | | | | |
|------|------|-----------|----------|----|----------|------|----|-------|--------|--|-------------------------------------|------|
| 58 | FF02 | CD | 1C | AE | FF00 | CD | 9A | 00528 | MOVZBL | LOCAL_ACE, 28(SP) | 1043 | |
| | | | 1C | AE | | 05 | C2 | 0052E | SUBL2 | #5, 28(SP) | | |
| | | | 1C | AE | | 04 | C6 | 00532 | DIVL2 | #4, 28(SP) | | |
| | | | | 04 | | 00 | EF | 00536 | EXTZV | #0, #4, LOCAL_ACE+2, J | | |
| | | | | | | 00B5 | 31 | 0053D | BRW | 54\$ | | |
| | | | 2C | AE | FF04 | CD48 | D0 | 00540 | 49\$: | MOVL | LOCAL_ACE+4[CJ], KEY_IDENTIFIER | 1046 |
| | | | 0A54 | CE | 0200 | 8F | B0 | 00547 | MOVW | #512, FAO_DESC | 1047 | |
| | | | 0A58 | CE | 0854 | CE | 9E | 0054E | MOVAB | FAO_BUF, FAO_DESC+4 | 1048 | |
| | | | | | 2C | AE | 9F | 00553 | PUSHAB | KEY_IDENTIFIER | 1052 | |
| | | | | | 0A58 | CE | 9F | 00558 | PUSHAB | FAO_DESC | | |
| | | | | | 0A5C | CE | 9F | 0055C | PUSHAB | FAO_DESC | | |
| | | | | | 00000000 | EF | 9F | 00560 | PUSHAB | P.ADI | | |
| | | 00000000G | 00 | | | 04 | FB | 00566 | CALLS | #4, SYSSFAOL | | |
| | | | 2F | | 28 | AE | E9 | 0056D | BLBC | 40(SP), 51\$ | 1053 | |
| | | | 50 | | 0A54 | CE | 3C | 00571 | MOVZWL | FAO_DESC, RO | | |
| | | | 50 | | | 57 | C0 | 00576 | ADDL2 | LINE_SIZE, RO | | |
| | | | 59 | | | 50 | D1 | 00579 | CMPL | RO, WIDTH | | |
| | | | | | | 22 | 1B | 0057C | BLEQU | 51\$ | | |
| | | | | | | 5B | D5 | 0057E | TSTL | TERM_LENGTH | | |
| | | | | | | 0F | 15 | 00580 | BLEQ | 50\$ | | |
| | 0A5C | CE46 | 08 | BE | | 5B | 28 | 00582 | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | | |
| | | | | 50 | 14 | BC | 3C | 0058A | MOVZWL | @TERM_DESC, RO | | |
| | | | | 56 | | 50 | C0 | 0058E | ADDL2 | RO, SIZE | | |
| 5A | | 20 | | 6E | | 00 | 2C | 00591 | 50\$: | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | | 0A5C | CE46 | | 00596 | | | | |
| | | | | 56 | | 5A | C0 | 0059A | ADDL2 | INDENT, SIZE | | |
| | | | | 57 | | 5A | D0 | 0059D | MOVL | INDENT, LINE_SIZE | | |
| | | | 1B | AE | 0A54 | CE | 3C | 005A0 | 51\$: | MOVZWL | FAO_DESC, 24(SP) | |
| | | | | 57 | 18 | AE | C0 | 005A6 | ADDL2 | 24(SP), LINE_SIZE | | |
| | 0A5C | CE46 | 0A58 | DE | 18 | AE | 28 | 005AA | MOV3 | 24(SP), @FAO_DESC+4, BUFFER[SIZE] | 1056 | |
| | | | | 56 | 18 | AE | C0 | 005B4 | ADDL2 | 24(SP), SIZE | 1057 | |
| | | | | 2B | 28 | AE | E9 | 005B8 | BLBC | 40(SP), 53\$ | 1058 | |
| | | | | 50 | 01 | A7 | 9E | 005BC | MOVAB | 1(R7), RO | | |
| | | | | 59 | | 50 | D1 | 005C0 | CMPL | RO, WIDTH | | |
| | | | | | | 22 | 1B | 005C3 | BLEQU | 53\$ | | |
| | | | | | | 5B | D5 | 005C5 | TSTL | TERM_LENGTH | | |
| | | | | | | 0F | 15 | 005C7 | BLEQ | 52\$ | | |
| | 0A5C | CE46 | 08 | BE | | 5B | 28 | 005C9 | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | | |
| | | | | 50 | 14 | BC | 3C | 005D1 | MOVZWL | @TERM_DESC, RO | | |
| | | | | 56 | | 50 | C0 | 005D5 | ADDL2 | RO, SIZE | | |
| 5A | | 20 | | 6E | | 00 | 2C | 005D8 | 52\$: | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | | 0A5C | CE46 | | 005DD | | | | |
| | | | | 56 | | 5A | C0 | 005E1 | ADDL2 | INDENT, SIZE | | |
| | | | | 57 | | 5A | D0 | 005E4 | MOVL | INDENT, LINE_SIZE | | |
| | | | | | | 57 | D6 | 005E7 | 53\$: | INCL | LINE_SIZE | |
| | 0A5C | CE46 | 00000000 | | | EF | 90 | 005E9 | MOV3 | P.ADR, BUFFER[SIZE] | | |
| | | | | | | 56 | D6 | 005F3 | INCL | SIZE | | |
| FF44 | | 58 | | 01 | 1C | AE | F1 | 005F5 | 54\$: | ACBL | 28(SP), #1, J 49\$ | 1043 |
| | | | | | 0A5B | CE46 | | 005FC | MOV3 | #44, BUFFER-1[SIZE] | 1060 | |
| | | | | | | 0718 | 31 | 00602 | BRW | 126\$ | 1037 | |
| | | | | 02 | 18 | AE | 91 | 00605 | 55\$: | CMPB | 24(SP), #2 | 1066 |
| | | | | | | 40 | 12 | 00609 | BNEQ | 58\$ | | |
| | | | | 2B | 28 | AE | E9 | 0060B | BLBC | 40(SP), 57\$ | 1067 | |
| | | | | 50 | 0B | A7 | 9E | 0060F | MOVAB | 11(R7), RO | | |
| | | | | 59 | | 50 | D1 | 00613 | CMPL | RO, WIDTH | | |
| | | | | | | 22 | 1B | 00616 | BLEQU | 57\$ | | |
| | | | | | | 5B | D5 | 00618 | TSTL | TERM_LENGTH | | |

| | | | | | | |
|---------------------|----|----|----------------------|--------|--|------|
| 0A5C CE46 | 08 | BE | OF 15 0061A | BLEQ | 56\$ | |
| | | 50 | 5B 28 0061C | MOV C3 | TERM LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 56 | 14 BC 3C 00624 | MOVZWL | @TERM_DESC, R0 | |
| SA | 20 | 6E | 50 C0 00628 | ADDL2 | R0, SIZE | |
| | | | 00 2C 0062B 56\$: | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 56 | 0A5C CE46 00630 | | | |
| | | 57 | 5A C0 00634 | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A D0 00637 | MOVL | INDENT, LINE SIZE | |
| 0A5C CE46 00000000' | | 57 | 0B C0 0063A 57\$: | ADDL2 | #11, LINE SIZE | |
| | | 56 | 0B 28 0063D | MOV C3 | #11, P.ADC, BUFFER[SIZE] | |
| | | 03 | 0B C0 00648 | ADDL2 | #11, SIZE | |
| | | | 18 AE 91 0064B 58\$: | CMPB | 24(SP), #3 | 1068 |
| | | 2B | 40 12 0064F | BNEQ | 61\$ | |
| | | 50 | 28 AE E9 00651 | BLBC | 40(SP), 60\$ | 1069 |
| | | 59 | 0B A7 9E 00655 | MOVAB | 11(R7), R0 | |
| | | | 50 D1 00659 | CMP L | R0, WIDTH | |
| | | | 22 1B 0065C | BLEQU | 60\$ | |
| | | | 5B D5 0065E | TSTL | TERM_LENGTH | |
| 0A5C CE46 | 08 | BE | OF 15 00660 | BLEQ | 59\$ | |
| | | 50 | 5B 28 00662 | MOV C3 | TERM LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 56 | 14 BC 3C 0066A | MOVZWL | @TERM_DESC, R0 | |
| SA | 20 | 6E | 50 C0 0066E | ADDL2 | R0, SIZE | |
| | | | 00 2C 00671 59\$: | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 56 | 0A5C CE46 00676 | | | |
| | | 57 | 5A C0 0067A | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A D0 0067D | MOVL | INDENT, LINE SIZE | |
| 0A5C CE46 00000000' | | 57 | 0B C0 00680 60\$: | ADDL2 | #11, LINE SIZE | |
| | | 56 | 0B 28 00683 | MOV C3 | #11, P.ADM, BUFFER[SIZE] | |
| | | 04 | 0B C0 0068E | ADDL2 | #11, SIZE | |
| | | | 18 AE 91 00691 61\$: | CMPB | 24(SP), #4 | 1070 |
| | | 2B | 40 12 00695 | BNEQ | 64\$ | |
| | | 50 | 28 AE E9 00697 | BLBC | 40(SP), 63\$ | 1071 |
| | | 59 | 0B A7 9E 0069B | MOVAB | 11(R7), R0 | |
| | | | 50 D1 0069F | CMP L | R0, WIDTH | |
| | | | 22 1B 006A2 | BLEQU | 63\$ | |
| | | | 5B D5 006A4 | TSTL | TERM_LENGTH | |
| 0A5C CE46 | 08 | BE | OF 15 006A6 | BLEQ | 62\$ | |
| | | 50 | 5B 28 006A8 | MOV C3 | TERM LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 56 | 14 BC 3C 006B0 | MOVZWL | @TERM_DESC, R0 | |
| SA | 20 | 6E | 50 C0 006B4 | ADDL2 | R0, SIZE | |
| | | | 00 2C 006B7 62\$: | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 56 | 0A5C CE46 006BC | | | |
| | | 57 | 5A C0 006C0 | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A D0 006C3 | MOVL | INDENT, LINE SIZE | |
| 0A5C CE46 00000000' | | 57 | 0B C0 006C6 63\$: | ADDL2 | #11, LINE SIZE | |
| | | 56 | 0B 28 006C9 | MOV C3 | #11, P.ADN, BUFFER[SIZE] | |
| | | 31 | 0B C0 006D4 | ADDL2 | #11, SIZE | |
| | | 50 | 28 AE E9 006D7 64\$: | BLBC | 40(SP), 66\$ | 1072 |
| | | 50 | FF00 CD 9A 006DB | MOVZBL | LOCAL ACE, R0 | |
| | | 59 | FC A047 9E 006E0 | MOVAB | -4(R0)[LINE_SIZE], R0 | |
| | | | 50 D1 006E5 | CMP L | R0, WIDTH | |
| | | | 22 1B 006E8 | BLEQU | 66\$ | |
| | | | 5B D5 006EA | TSTL | TERM_LENGTH | |
| 0A5C CE46 | 08 | BE | OF 15 006EC | BLEQ | 65\$ | |
| | | 50 | 5B 28 006EE | MOV C3 | TERM LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 56 | 14 BC 3C 006F6 | MOVZWL | @TERM_DESC, R0 | |
| | | | 50 C0 006FA | ADDL2 | R0, SIZE | |

| | | | | | | | | | |
|----|------|------|-----------|------|-------|-------|-------|-------------------------------------|--|
| 5A | 20 | 6E | 00 | 2C | 006FD | 65\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 56 | 0A5C | CE46 | 00702 | | | | |
| | | 57 | 5A | C0 | 00706 | | ADDL2 | INDENT, SIZE | |
| | | 58 | 5A | D0 | 00709 | | MOVL | INDENT, LINE_SIZE | |
| | | 57 | FF00 | CD | 9A | 0070C | 66\$: | MOVZBL | LOCAL ACE, R8 |
| | | 50 | FC | A847 | 9E | 00711 | | MOVAB | -4(R8)[LINE_SIZE], LINE_SIZE |
| | | 56 | FC | A8 | 9E | 00716 | | MOVAB | -4(R8), R0 |
| | 0A5C | CE46 | FF04 | CD | 50 | 28 | 0071A | MOVCS | R0, LOCAL ACE+4, BUFFER[SIZE] |
| | | 56 | FC | A846 | 9E | 00723 | | MOVAB | -4(R8)[SIZE], SIZE |
| | | 28 | 28 | AE | E9 | 00728 | | BLBC | 40(SP), 68\$ |
| | | 50 | 01 | A7 | 9E | 0072C | | MOVAB | 1(R7), R0 |
| | | 59 | 50 | D1 | 00730 | | CMPL | R0, WIDTH | |
| | | | 22 | 1B | 00733 | | BLEQU | 68\$ | |
| | | | 5B | D5 | 00735 | | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | 00737 | | BLEQ | 67\$ | |
| | 0A5C | CE46 | 08 | 8E | 5B | 28 | 00739 | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] |
| | | 50 | 14 | BC | 3C | 00741 | | MOVZWL | @TERM_DESC, R0 |
| | | 56 | 50 | C0 | 00745 | | ADDL2 | R0, SIZE | |
| 5A | 20 | 6E | 00 | 2C | 00748 | 67\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 56 | 0A5C | CE46 | 0074D | | | | |
| | | 57 | 5A | C0 | 00751 | | ADDL2 | INDENT, SIZE | |
| | | | 5A | D0 | 00754 | | MOVL | INDENT, LINE_SIZE | |
| | | | 57 | D6 | 00757 | 68\$: | INCL | LINE_SIZE | |
| | 0A5C | CE46 | 00000000' | EF | 90 | 00759 | | MOVAB | P.ADD, BUFFER[SIZE] |
| | | | 0120 | 31 | 00763 | | BRW | 80\$ | |
| | 20 | AE | 01 | 90 | 00766 | 69\$: | MOVAB | #1, ACCESS MASK | 1082 |
| | 24 | AE | 01 | 90 | 0076A | | MOVAB | #1, AUDIT MASK | 1083 |
| | | 05 | 18 | AE | 91 | 0076E | | CMPB | 24(SP), #5 |
| | | | 40 | 12 | 00772 | | BNEQ | 72\$ | 1084 |
| | | 28 | 28 | AE | E9 | 00774 | | BLBC | 40(SP), 71\$ |
| | | 50 | 0E | A7 | 9E | 00778 | | MOVAB | 14(R7), R0 |
| | | 59 | 50 | D1 | 0077C | | CMPL | R0, WIDTH | 1085 |
| | | | 22 | 1B | 0077F | | BLEQU | 71\$ | |
| | | | 5B | D5 | 00781 | | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | 00783 | | BLEQ | 70\$ | |
| | 0A5C | CE46 | 08 | 8E | 5B | 28 | 00785 | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] |
| | | 50 | 14 | BC | 3C | 0078D | | MOVZWL | @TERM_DESC, R0 |
| | | 56 | 50 | C0 | 00791 | | ADDL2 | R0, SIZE | |
| 5A | 20 | 6E | 00 | 2C | 00794 | 70\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | 0A5C | CE46 | 00799 | | | | |
| | | 56 | 5A | C0 | 0079D | | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A | D0 | 007A0 | | MOVL | INDENT, LINE_SIZE | |
| | | 57 | 0E | C0 | 007A3 | 71\$: | ADDL2 | #14, LINE_SIZE | |
| | 0A5C | CE46 | 00000000' | EF | 0E | 28 | 007A6 | MOVCS | #14, P.ADD, BUFFER[SIZE] |
| | | 56 | 0E | C0 | 007B1 | | ADDL2 | #14, SIZE | |
| | | 06 | 18 | AE | 91 | 007B4 | 72\$: | CMPB | 24(SP), #6 |
| | | | 40 | 12 | 007B8 | | BNEQ | 75\$ | 1086 |
| | | 28 | 28 | AE | E9 | 007BA | | BLBC | 40(SP), 74\$ |
| | | 50 | 0E | A7 | 9E | 007BE | | MOVAB | 14(R7), R0 |
| | | 59 | 50 | D1 | 007C2 | | CMPL | R0, WIDTH | 1087 |
| | | | 22 | 1B | 007C5 | | BLEQU | 74\$ | |
| | | | 5B | D5 | 007C7 | | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | 007C9 | | BLEQ | 73\$ | |
| | 0A5C | CE46 | 08 | 8E | 5B | 28 | 007CB | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] |
| | | 50 | 14 | BC | 3C | 007D3 | | MOVZWL | @TERM_DESC, R0 |
| | | 56 | 50 | C0 | 007D7 | | ADDL2 | R0, SIZE | |
| 5A | 20 | 6E | 00 | 2C | 007DA | 73\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |

| Address | Disassembly | Comment | Symbol |
|-------------------------------|-----------------|---------|--|
| 0A5C CE46 00000000' | 56 5A C0 007DF | ADDL2 | INDENT, SIZE |
| | 57 5A D0 007E3 | MOVL | INDENT, LINE_SIZE |
| | 57 0E C0 007E9 | ADDL2 | #14, LINE_SIZE |
| | EF 0E 28 007EC | MOVC3 | #14, P.ADR, BUFFER[SIZE] |
| | 56 0E C0 007F7 | ADDL2 | #14, SIZE |
| 31 28 AE E9 007FA | 75\$: | BLBC | 40(SP), 77\$ |
| 50 FF00 CD 9A 007FE | | MOVZBL | LOCAL ACE, R0 |
| 50 FB A047 9E 00803 | | MOVAB | -8(R0)[LINE_SIZE], R0 |
| 59 50 D1 00808 | | CMPL | R0, WIDTH |
| | 22 1B 0080B | BLEQU | 77\$ |
| | 5B D5 0080D | TSTL | TERM_LENGTH |
| | OF 15 0080F | BLEQ | 76\$ |
| 0A5C CE46 DB BE 5B 28 00811 | | MOVC3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] |
| 50 14 BC 3C 00819 | | MOVZWL | @TERM_DESC, R0 |
| 56 50 C0 0081D | | ADDL2 | R0, SIZE |
| 6E 00 2C 00820 | 76\$: | MOVC5 | #0, (SP), #32, INDENT, BUFFER[SIZE] |
| | 0A5C CE46 00825 | ADDL2 | INDENT, SIZE |
| 56 5A C0 00829 | | MOVL | INDENT, LINE_SIZE |
| 57 5A D0 0082C | | MOVZBL | LOCAL ACE, R8 |
| 58 FF00 CD 9A 0082F | 77\$: | MOVAB | -8(R8)[LINE_SIZE], LINE_SIZE |
| 57 FB A847 9E 00834 | | MOVAB | -8(R8), R0 |
| 50 FB A8 9E 00839 | | MOVC3 | R0, LOCAL ACE+8, BUFFER[SIZE] |
| 0A5C CE46 FF08 CD 50 28 0083D | | MOVAB | -8(R8)[SIZE], SIZE |
| 56 FB A846 9E 00846 | | BLBC | 40(SP), 79\$ |
| 2B 28 AE E9 0084B | | MOVAB | 1(R7), R0 |
| 50 01 A7 9E 0084F | | CMPL | R0, WIDTH |
| 59 50 D1 00853 | | BLEQU | 79\$ |
| | 22 1B 00856 | TSTL | TERM_LENGTH |
| | 5B D5 00858 | BLEQ | 78\$ |
| 0A5C CE46 DB BE 5B 28 0085C | | MOVC3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] |
| 50 14 BC 3C 00864 | | MOVZWL | @TERM_DESC, R0 |
| 56 50 C0 00868 | | ADDL2 | R0, SIZE |
| 6E 00 2C 0086B | 78\$: | MOVC5 | #0, (SP), #32, INDENT, BUFFER[SIZE] |
| | 0A5C CE46 00870 | ADDL2 | INDENT, SIZE |
| 56 5A C0 00874 | | MOVL | INDENT, LINE_SIZE |
| 57 5A D0 00877 | | INCL | LINE_SIZE |
| 0A5C CE46 00000000' | 57 D6 0087A | MOVB | P.ADR, BUFFER[SIZE] |
| | EF 90 0087C | INCL | SIZE |
| | 56 D6 00886 | BRW | 126\$ |
| 2B 28 0492 31 00888 | 80\$: | BLBC | 40(SP), 83\$ |
| 50 13 A7 9E 0088F | 81\$: | MOVAB | 19(R7), R0 |
| 59 50 D1 00893 | | CMPL | R0, WIDTH |
| | 22 1B 00896 | BLEQU | 83\$ |
| | 5B D5 00898 | TSTL | TERM_LENGTH |
| | OF 15 0089A | BLEQ | 82\$ |
| 0A5C CE46 DB BE 5B 28 0089C | | MOVC3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] |
| 50 14 BC 3C 008A4 | | MOVZWL | @TERM_DESC, R0 |
| 56 50 C0 008AB | | ADDL2 | R0, SIZE |
| 6E 00 2C 008AB | 82\$: | MOVC5 | #0, (SP), #32, INDENT, BUFFER[SIZE] |
| | 0A5C CE46 008B0 | ADDL2 | INDENT, SIZE |
| 56 5A C0 008B4 | | MOVL | INDENT, LINE_SIZE |
| 57 5A D0 008B7 | | ADDL2 | #19, LINE_SIZE |
| 0A5C CE46 00000000' | 57 13 C0 008BA | MOVC3 | #19, P.ADR, BUFFER[SIZE] |
| | EF 13 28 008BD | | |

| | | | | | | | | | | |
|------|------------|----------------------------------|----------|------------------------|--|---------------------------------|---|---|--|------------------------------|
| 0038 | 03 0028 | 56 00 0018 | 1C 1C | 13 AE AE 0008 | CO D4 CF 008C8 008CB 008CE 84\$: 008D3 85\$: | ADDL2 CLRL CASEL .WORD | #19, SIZE K K, #0, #3 86\$-85\$,- 87\$-85\$,- 88\$-85\$,- 89\$-85\$ | 1098 1101 | | |
| | | 18 04 | AE AE | FF08 00000000' | CD EF | D0 9E | 008DB 86\$: 008E1 | MOVL MOVAB | LOCAL_ACE+8, PROT_VALUE P.ADT, PROT_FIELD_DSC | 1104 1105 |
| | | 18 04 | AE AE | FF0C 00000000' | CD EF | D0 9E | 008EB 87\$: 008F1 | MOVL MOVAB | LOCAL_ACE+12, PROT_VALUE P.ADV, PROT_FIELD_DSC | 1101 1108 1109 |
| | | 18 04 | AE AE | FF10 00000000' | CD EF | D0 9E | 008FB 88\$: 00901 | MOVL MOVAB | LOCAL_ACE+16, PROT_VALUE P.ADX, PROT_FIELD_DSC | 1101 1112 1113 |
| | | 18 04 | AE AE | FF14 00000000' | CD EF | D0 9E | 00909 89\$: 0090B 89\$: 00911 | MOVL MOVAB | LOCAL_ACE+20, PROT_VALUE P.ADZ, PROT_FIELD_DSC | 1101 1116 1117 |
| | | | | 51 00000000'EF | 58 50 40 | D4 D4 9A | 00919 90\$: 0091B 0091D 91\$: | CLRL CLRL MOVZBL | PROT_IDX J PROT_CODE[J], R1 | 1120 1121 1124 |
| | 08 | 18 FED8 | AE CD | | 50 51 | E0 90 | 00925 00927 0092C | BEQL BBS MOVAB | J, PROT_VALUE, 92\$ R1, PROT_BUF[PROT_IDX] | |
| | E5 | 50 31 50 50 50 59 | | | 58 1F 31 57 58 | D6 F3 AE C0 C0 | 00932 00934 92\$: 00938 0093C 00940 00943 | INCL AOBLEQ BLBC MOVZWL ADDL2 | PROT_IDX #31, J, 91\$ 40(SP), 94\$ @PROT_FIELD_DSC, R0 LINE_SIZE, R0 | 1127 1128 1121 1131 |
| | | | | | 50 22 58 OF | D1 1B D5 15 | 00946 00949 0094B 0094D | CMP BLEQU TSTL | R0, WIDTH 94\$ TERM_LENGTH | |
| | 0A5C CE46 | 08 | BE | | 5B | 28 | 0094F | BLEQ | 93\$ | |
| | | | | | 5B | 3C | 00957 | MCVC3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | SA | 20 | 56 6E | | 14 50 | BC C0 | 00957 0095B | MOVZWL ADDL2 | @TERM_DESC, R0 R0, SIZE | |
| | | | | | 00 | 2C | 0095E 93\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | | 56 57 | SA SA | 00963 00967 0096A | ADDL2 MOVL | INDENT, SIZE INDENT, LINE_SIZE | |
| | | 10 | AE | | 04 10 | BE AE | 0096D 94\$: 00972 | MOVZWL ADDL3 | @PROT_FIELD_DSC, 16(SP) 16(SP), LINE_SIZE, R0 | |
| | | 50 57 | AE | | 58 56 | C1 C3 | 00977 0097B | ADDL3 SUBL3 | PROT_IDX, R0, LINE_SIZE SIZE, #512, 12(SP) | |
| | OC | AE 00000200 | 8F | | 56 | C3 | 0097B | MOVAB | BUFFER[SIZE], 20(SP) | 1135 1136 |
| | | 7E | 04 | AE | 0A5C CE46 | 9E | 00984 | ADDL3 | #4, PROT_FIELD_DSC, -(SP) | |
| | | | | | 04 | C1 | 0098B | PUSHL | @(SP)+ | |
| | OC | AE | 00 | 9E | 14 | AE | 00990 | MOVCS | 20(SP), @(SP)+, #0, 12(SP), @20(SP) | |
| | | | | | 14 | BE | 00999 | | | |
| | | | | | 14 | 18 | 0099B | BGEQ | 95\$ | |
| | | | | | 10 | AE | 0099D | ADDL2 | 16(SP), 20(SP) | |
| | | | | | 10 | AE | 009A2 | SUBL2 | 16(SP), 12(SP) | |
| | OC | AE | 00 | FED8 | CD | 58 | 009A7 | MOVCS | PROT_IDX, PROT_BUF, #0, 12(SP), @20(SP) | |
| | | | | | 14 | BE | 009AF | | | |
| | | 50 56 | 56 50 | | 10 58 | AE C1 | 009B1 95\$: 009B6 | ADDL3 ADDL3 | 16(SP), SIZE, R0 PROT_IDX, R0, SIZE | 1137 |

| | | | | | | | | | | |
|------|------|------|-----------|------|------|-------|-------|---------------------|--|--|
| | | 2B | 28 | AE | E9 | 009BA | BLBC | 40(SP), 97\$ | 1138 | |
| | | 50 | 01 | A7 | 9E | 009BE | MOVAB | 1(R7), R0 | | |
| | | 59 | | 50 | D1 | 009C2 | CMPL | R0, WIDTH | | |
| | | | | 22 | 1B | 009C5 | BLEQU | 97\$ | | |
| | | | | 5B | D5 | 009C7 | TSTL | TERM_LENGTH | | |
| | | | | 0F | 15 | 009C9 | BLEQ | 96\$ | | |
| | 0A5C | CE46 | 08 | BE | 5B | 28 | 009CB | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | | 50 | 14 | BC | 3C | 009D3 | | |
| | | | | 56 | 50 | C0 | 009D7 | MOVZWL | @TERM_DESC, R0 | |
| SA | | 20 | | 6E | 00 | 2C | 009DA | ADDL2 | R0, SIZE | |
| | | | | | 0A5C | CE46 | 009DF | MOV3 | R0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | 56 | 5A | C0 | 009E3 | ADDL2 | INDENT, SIZE | |
| | | | | 57 | 5A | D0 | 009E6 | MOVL | INDENT, LINE_SIZE | |
| | | | | | 57 | D6 | 009E9 | INCL | LINE_SIZE | |
| | 0A5C | CE46 | 00000000' | EF | 90 | 009EB | MOV3 | P.AE3, BUFFER[SIZE] | | |
| | | | | 56 | D6 | 009F5 | INCL | SIZE | | |
| FED0 | 1C | AE | | 01 | 03 | F1 | 009F7 | ACBL | #3, #1, K, 84\$ | |
| | | | | | 031C | 31 | 009FE | BRW | 126\$ | |
| | | | | 2B | 28 | AE | E9 | 00A01 | 98\$: | |
| | | | | 50 | 0F | A7 | 9E | 00A05 | BLBC | 40(SP), 100\$ |
| | | | | 59 | | 50 | D1 | 00A09 | MOVAB | 15(R7), R0 |
| | | | | | | 22 | 1B | 00A0C | CMPL | R0, WIDTH |
| | | | | | | 5B | D5 | 00A0E | BLEQU | 100\$ |
| | | | | | | 0F | 15 | 00A10 | TSTL | TERM_LENGTH |
| | 0A5C | CE46 | 08 | BE | 5B | 28 | 00A12 | BLEQ | 99\$ | |
| | | | | 50 | 14 | BC | 3C | 00A1A | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] |
| | | | | 56 | | 50 | C0 | 00A1E | MOVZWL | @TERM_DESC, R0 |
| SA | | 20 | | 6E | 00 | 2C | 00A21 | ADDL2 | R0, SIZE | |
| | | | | | 0A5C | CE46 | 00A26 | MOV3 | R0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | 56 | 5A | C0 | 00A2A | ADDL2 | INDENT, SIZE | |
| | | | | 57 | 5A | D0 | 00A2D | MOVL | INDENT, LINE_SIZE | |
| | | | | 57 | 0F | C0 | 00A30 | ADDL2 | #15, LINE_SIZE | |
| | 0A5C | CE46 | 00000000' | EF | 0F | 28 | 00A33 | MOV3 | #15, P.AEC, BUFFER[SIZE] | |
| | | | | 56 | 0F | C0 | 00A3E | ADDL2 | #15, SIZE | |
| 08 | | 00 | | 6E | 00 | 2C | 00A41 | MOV3 | R0, (SP), #0, #8, VOLNAM_DESC | |
| | | | | | 084C | CE | 00A46 | | | |
| 12 | | 00 | | 6E | 00 | 2C | 00A49 | MOV3 | R0, (SP), #0, #18, VOLNAM_TEXT | |
| | | | | | 0838 | CE | 00A4E | | | |
| 08 | | 00 | | 6E | 00 | 2C | 00A51 | MOV3 | R0, (SP), #0, #8, FILENAME_DESC | |
| | | | | | 0830 | CE | 00A56 | | | |
| 0200 | 8F | 00 | | 6E | 00 | 2C | 00A59 | MOV3 | R0, (SP), #0, #512, FILENAME_TEXT | |
| | | | | | 30 | AE | 00A60 | | | |
| | 0838 | CE | 00000000' | EF | 05 | 28 | 00A62 | MOV3 | #5, P.AED, VOLNAM_TEXT | |
| 0D | | 00 | FF04 | CD | 0C | 2C | 00A6C | MOV3 | #12, LOCAL_ACE+4, #0, #13, (R3) | |
| | | | | | 63 | | 00A73 | | | |
| | 0838 | CE | | 12 | 00 | 3A | 00A74 | LOCC | #0, #18, VOLNAM_TEXT | |
| | | | | | 02 | 12 | 00A7A | BNEQ | 101\$ | |
| | | | | | 51 | D4 | 00A7C | CLRL | R1 | |
| | | | | 50 | 0838 | CE | 9E | 00A7E | MOVAB | VOLNAM_TEXT, R0 |
| | 084C | CE | | 51 | 50 | A3 | 00A83 | SUBW3 | R0, R1, VOLNAM_DESC | |
| | | 0850 | | CE | 0838 | CE | 9E | 00A89 | MOVAB | VOLNAM_TEXT, VOLNAM_DESC+4 |
| | | 0830 | | CE | 0200 | 8F | B0 | 00A90 | MOVW | #512, FILENAME_DESC |
| | | 0834 | | CE | 30 | AE | 9E | 00A97 | MOVAB | FILENAME_TEXT, FILENAME_DESC+4 |
| | | | | 0830 | CE | 9F | 00A9D | PUSHAB | FILENAME_DESC | |
| | | | | 0834 | CE | 9F | 00AA1 | PUSHAB | FILENAME_DESC | |
| | | | | FF10 | CD | 9F | 00AA5 | PUSHAB | LOCAL_ACE+16 | |
| | | | | 0858 | CE | 9F | 00AA9 | PUSHAB | VOLNAM_DESC | |

| | | | | | | | |
|---------------------|------|-----------|-----------|-------|--------|--|------|
| 00000000G | 00 | 04 | FB | 00AAD | CALLS | #4, LIB\$FID TO NAME | |
| | 58 | 50 | D0 | 00AB4 | MOVL | R0, LOCAL STATUS | |
| | 28 | 28 | AE | E9 | BLBC | 40(SP), 103\$ | 1158 |
| | 50 | OF | A7 | 9E | MOVAB | 15(R7), R0 | |
| | 59 | | 50 | D1 | CMPL | R0, WIDTH | |
| | | | 22 | 1B | BLEQU | 103\$ | |
| | | | 5B | D5 | TSTL | TERM_LENGTH | |
| | | | OF | 15 | BLEQ | 102\$ | |
| 0A5C CE46 | 08 | BE | 5B | 28 | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | 50 | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | 56 | | 50 | C0 | ADDL2 | R0, SIZE | |
| SA | 20 | 6E | 00 | 2C | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 0A5C CE46 | 2C | 00AD7 | | | |
| | | | 5A | 00ADC | ADDL2 | INDENT, SIZE | |
| | | | 5A | D0 | MOVL | INDENT, LINE_SIZE | |
| | | | OF | C0 | ADDL2 | #15, LINE_SIZE | |
| 0A5C CE46 00000000' | | 57 | OF | 28 | MOVCS | #15, P.AEE, BUFFER[SIZE] | |
| | | 56 | OF | C0 | ADDL2 | #15, SIZE | |
| | | 03 | 58 | E8 | BLBS | LOCAL_STATUS, 104\$ | 1159 |
| | | | 00E8 | 31 | BRW | 117\$ | |
| | 18 | AE | 0834 | CE | MOVL | FILENAME_DESC+4, SEGMENT_START | 1164 |
| | 59 | 10 | 57 | C3 | SUBL3 | LINE_SIZE, WIDTH, R0 | 1165 |
| 50 | 0830 | 50 | 00 | ED | CMPZV | #0, #16, FILENAME_DESC, R0 | |
| | CE | | 05 | 1E | BGEQU | 105\$ | |
| | | 50 | 0830 | CE | MOVZWL | FILENAME_DESC, R0 | |
| | | 58 | 50 | D0 | MOVL | R0, SEGMENT_SIZE | |
| | 1C | AE | 0830 | CE | MOVZWL | FILENAME_DESC, 28(SP) | 1168 |
| | 1C | AE | 58 | D1 | CMPL | SEGMENT_SIZE, 28(SP) | |
| | | | 2C | 1E | BGEQU | 110\$ | |
| | 50 | | 01 | A8 | MOVAB | 1(R8), J | 1173 |
| | | | 23 | 11 | BRB | 109\$ | |
| | 52 | | 18 | AE | MOVL | SEGMENT_START, R2 | |
| | 51 | | FF | A042 | MOVZBL | -1(J)[R2], R1 | |
| | 3A | | 51 | 91 | CMPB | R1, #58 | |
| | | | 10 | 13 | BEQL | 108\$ | |
| 50 | 8F | | 51 | 91 | CMPB | R1, #93 | 1174 |
| | | | 0A | 13 | BEQL | 108\$ | |
| | 2E | | 51 | 91 | CMPB | R1, #46 | 1175 |
| | | | 05 | 13 | BEQL | 108\$ | |
| | 3B | | 51 | 91 | CMPB | R1, #59 | 1176 |
| | | | 05 | 12 | BNEQ | 109\$ | |
| | 58 | | 50 | D0 | MOVL | J, SEGMENT_SIZE | 1179 |
| | | | 03 | 11 | BRB | 110\$ | 1178 |
| | | | 50 | F5 | SOBGR | J, 107\$ | 1170 |
| 0A5C CE46 | 1B | DA | 58 | 28 | MOVCS | SEGMENT_SIZE, @SEGMENT_START, BUFFER[SIZE] | 1183 |
| | | BE | 58 | C0 | ADDL2 | SEGMENT_SIZE, LINE_SIZE | 1184 |
| | | 57 | 58 | C0 | ADDL2 | SEGMENT_SIZE, SIZE | 1185 |
| | | 56 | 58 | A2 | SUBW2 | SEGMENT_SIZE, FILENAME_DESC | 1186 |
| | 0830 | CE | 58 | C0 | ADDL2 | SEGMENT_SIZE, SEGMENT_START | 1187 |
| | 18 | AE | 0830 | CE | MOVZWL | FILENAME_DESC, 28(SP) | 1188 |
| | 1C | AE | 1E | 15 | BLEQ | 112\$ | |
| | | | 5B | D5 | TSTL | TERM_LENGTH | |
| | | | 0B | 15 | BLEQ | 111\$ | |
| 0A5C CE46 | 0B | BE | 5B | 28 | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 56 | 5B | C0 | ADDL2 | TERM_LENGTH, SIZE | |
| SA | 20 | 6E | 00 | 2C | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | 0A5C CE46 | 00B83 | | | |

| | | | | | | | | | |
|-----------|----------|------|-----------|----|-------|--------|--|----------------|--------|
| | | 56 | SA | C0 | 00B87 | ADDL2 | INDENT, SIZE | | |
| | | 57 | SA | D0 | 00B8A | MOVL | INDENT, LINE_SIZE | | |
| 50 | | 59 | 57 | C3 | 00B8D | SUBL3 | LINE_SIZE, WIDTH, R0 | 1189 | |
| | 1C | AE | 50 | D1 | 00B91 | CMPL | R0, 28(SP) | | |
| | | | 04 | 1B | 00B95 | BLEQU | 113\$ | | |
| | | 50 | 1C | AE | D0 | 00B97 | MOVL | 28(SP), R0 | |
| | | 58 | 50 | D0 | 00B9B | MOVL | R0, SEGMENT_SIZE | | |
| | | | 1C | AE | D5 | 00B9E | TSTL | 28(SP) | 1191 |
| | | | 03 | 15 | 00BA1 | BLEQ | 114\$ | | |
| | | | FF78 | 31 | 00BA3 | BRW | 106\$ | | |
| | | 2B | 28 | AE | E9 | 00BA6 | BLBC | 40(SP), 116\$ | 1192 |
| | | 50 | 01 | A7 | 9E | 00BAA | MOVAB | 1(R7), R0 | |
| | | 59 | 50 | D1 | 00BAE | CMPL | R0, WIDTH | | |
| | | | 22 | 1B | 00BB1 | BLEQU | 116\$ | | |
| | | | 5B | D5 | 00BB3 | TSTL | TERM_LENGTH | | |
| | | | 0F | 15 | 00BB5 | BLEQ | 115\$ | | |
| 0A5C CE46 | 08 | BE | 5B | 28 | 00BB7 | MOVC3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | | |
| | | 50 | 14 | BC | 3C | 00BBF | MOVZWL | @TERM_DESC, R0 | |
| | | 56 | 50 | C0 | 00BC3 | ADDL2 | R0, SIZE | | |
| 5A | 20 | 6E | 00 | 2C | 00BC6 | MOVC5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | 115\$: | |
| | | | 0A5C CE46 | | 00BC8 | | | | |
| | | 56 | SA | C0 | 00BCF | ADDL2 | INDENT, SIZE | | |
| | | 57 | SA | D0 | 00BD2 | MOVL | INDENT, LINE_SIZE | | |
| | | | 57 | D6 | 00BD5 | INCL | LINE_SIZE | 116\$: | |
| 0A5C CE46 | 00000000 | | EF | 90 | 00BD7 | MOVAB | P.AEF, BUFFER[SIZE] | | |
| | | | 56 | D6 | 00BE1 | INCL | SIZE | | |
| | | | 76 | 11 | 00BE3 | BRB | 120\$ | 1159 | |
| 0A54 CE | 0200 | | 8F | B0 | 00BE5 | MOVW | #512, FAO_DESC | 1196 | |
| 0A58 CE | 0854 | | CE | 9E | 00BEC | MOVAB | FAO_BUF, FAO_DESC+4 | 1197 | |
| | FF14 | | CD | DD | 00BF3 | PUSHL | LOCAL_ACE+20 | 1203 | |
| | FF12 | | CD | DD | 00BF7 | PUSHL | LOCAL_ACE+18 | | |
| | FF10 | | CD | DD | 00BF8 | PUSHL | LOCAL_ACE+16 | | |
| | 0A60 | | CE | 9F | 00BFF | PUSHAB | FAO_DESC | | |
| | 0A64 | | CE | 9F | 00C03 | PUSHAB | FAO_DESC | | |
| | 00000000 | | EF | 9F | 00C07 | PUSHAB | P.AEG | | |
| 00000000G | 00 | | 06 | FB | 00C0D | CALLS | #6, SYSSFAO | | |
| | 2F | 28 | AE | E9 | 00C14 | BLBC | 40(SP), 119\$ | 1204 | |
| | 50 | 0A54 | CE | 3C | 00C18 | MOVZWL | FAO_DESC, R0 | | |
| | 50 | | 57 | C0 | 00C1D | ADDL2 | LINE_SIZE, R0 | | |
| | 59 | | 50 | D1 | 00C20 | CMPL | R0, WIDTH | | |
| | | | 22 | 1B | 00C23 | BLEQU | 119\$ | | |
| | | | 5B | D5 | 00C25 | TSTL | TERM_LENGTH | | |
| | | | 0F | 15 | 00C27 | BLEQ | 118\$ | | |
| 0A5C CE46 | 08 | BE | 5B | 28 | 00C29 | MOVC3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | | |
| | | 50 | 14 | BC | 3C | 00C31 | MOVZWL | @TERM_DESC, R0 | |
| | | 56 | 50 | C0 | 00C35 | ADDL2 | R0, SIZE | | |
| 5A | 20 | 6E | 00 | 2C | 00C38 | MOVC5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | 118\$: | |
| | | | 0A5C CE46 | | 00C3D | | | | |
| | | 56 | SA | C0 | 00C41 | ADDL2 | INDENT, SIZE | | |
| | | 57 | SA | D0 | 00C44 | MOVL | INDENT, LINE_SIZE | | |
| | | 58 | 0A54 | CE | 3C | 00C47 | MOVZWL | FAO_DESC, R8 | 119\$: |
| | | 57 | 58 | C0 | 00C4C | ADDL2 | R8, LINE_SIZE | | |
| 0A5C CE46 | 0A58 | DE | 58 | 28 | 00C4F | MOVC3 | R8, @FAO_DESC+4, BUFFER[SIZE] | | 1205 |
| | | 56 | 58 | C0 | 00C58 | ADDL2 | R8, SIZE | 1206 | |
| | | 2B | 28 | AE | E9 | 00C5B | BLBC | 40(SP), 122\$ | 1208 |
| | | 50 | 16 | A7 | 9E | 00C5F | MOVAB | 22(R7), R0 | |
| | | 59 | 50 | D1 | 00C63 | CMPL | R0, WIDTH | | |

| | | | | | | | | |
|----|--------------------|------|----|-------------|-------------|-------------|--|------|
| 5A | 0A5C CE46 | 08 | BE | 14 | 22 1B 00C66 | BLEQU | 122\$ | |
| | | | 50 | | 5B D5 00C68 | TSTL | TERM_LENGTH | |
| | | | 56 | | OF 15 00C6A | BLEQ | 121\$ | |
| | | | 6E | | 5B 28 00C6C | MOV C3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | | | BC 3C 00C74 | MOVZWL | @TERM_DESC, R0 | |
| | | | | | 50 C0 00C78 | ADDL2 | R0, SIZE | |
| | | | | | 00 2C 00C7B | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | 0A5C CE46 | 00C80 | | | |
| | | | 56 | 5A | C0 00C84 | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 00C87 | MOVL | INDENT, LINE_SIZE | |
| | | | 57 | 16 | C0 00C8A | ADDL2 | #22, LINE_SIZE | |
| | 0A5C CE46 00000000 | | EF | 16 | 28 00C8D | MOV C3 | #22, P.AET, BUFFER[SIZE] | |
| | | | 56 | 16 | C0 00C98 | ADDL2 | #22, SIZE | |
| | 0A54 CE | | | 8F | B0 00C9B | MOVW | #512, FAO_DESC | 1209 |
| | 0A58 CE | | | CE | 9E 00CA2 | MOVAB | FAO_BUF, FAO_DESC+4 | 1210 |
| | | | | FF18 | CD 9F 00CA9 | PUSHAB | LOCAL_ACE+24 | 1214 |
| | | | | 0A58 CE | 9F 00CAD | PUSHAB | FAO_DESC | |
| | | | | 0A5C CE | 9F 00CB1 | PUSHAB | FAO_DESC | |
| | | | | 00000000 | EF 9F 00CB5 | PUSHAB | P.AET | |
| | | | | 04 | F8 00CB8 | CALLS | #4, SYSSFAO | |
| | 00000000G | | 00 | 3A | 90 00CC2 | MOV B | #58, FAO_BUF+11 | 1215 |
| | 085F | | 20 | 0854 | CE 91 00CC7 | CMPB | FAO_BUF, -#32 | 1216 |
| | | | | 08 | 12 00CCC | BNEQ | 123\$ | |
| | | | | 0A54 CE | B7 00CCE | DECW | FAO_DESC | 1219 |
| | | | | 0A58 CE | D6 00CD2 | INCL | FAO_DESC+4 | 1220 |
| | | | 2F | AE | E9 00CD6 | BLBC | 40(SP), 125\$ | 1222 |
| | | | 50 | 0A54 | CE 3C 00CDA | MOVZWL | FAO_DESC, R0 | |
| | | | 50 | 57 | C0 00CDF | ADDL2 | LINE_SIZE, R0 | |
| | | | 59 | 50 | D1 00CE2 | CMPL | R0, WIDTH | |
| | | | | 22 1B 00CE5 | BLEQU | 125\$ | | |
| | | | | 5B D5 00CE7 | TSTL | TERM_LENGTH | | |
| | | | | OF 15 00CE9 | BLEQ | 124\$ | | |
| | 0A5C CE46 | 08 | BE | 14 | 5B 28 00CEB | MOV C3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 50 | BC | 3C 00CF3 | MOVZWL | @TERM_DESC, R0 | |
| | | | 56 | 50 | C0 00CF7 | ADDL2 | R0, SIZE | |
| | | | 6E | 00 | 2C 00CFA | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | | 0A5C CE46 | 00CFF | | | |
| | | | 56 | 5A | C0 00D03 | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 00D06 | MOVL | INDENT, LINE_SIZE | |
| | | | 58 | CE | 3C 00D09 | MOVZWL | FAO_DESC, R8 | |
| | | | 57 | 58 | C0 00D0E | ADDL2 | R8, LINE_SIZE | |
| | 0A5C CE46 | 0A58 | DE | 58 | 28 00D11 | MOV C3 | R8, @FAO_DESC+4, BUFFER[SIZE] | 1223 |
| | | | 56 | 58 | C0 00D1A | ADDL2 | R8, SIZE | 1224 |
| | | | 58 | FF02 | CD 90 00D1D | MOVW | LOCAL_ACE+2, FLAGS | 1293 |
| | | | 03 | 24 | AE E9 00D22 | BLBC | AUDIT_MASK, 127\$ | 1294 |
| | | | 58 | 03 | 8A 00D26 | BICB2 | #3, FLAGS | 1295 |
| | | | | 58 | B5 00D29 | TSTW | FLAGS | 1296 |
| | | | | 03 | 12 00D2B | BNEQ | 128\$ | |
| | | | | 01B6 | 31 00D2D | BRW | 146\$ | |
| | | | 28 | AE | E9 00D30 | BLBC | 40(SP), 130\$ | 1299 |
| | | | 50 | 08 | A7 9E 00D34 | MOVAB | 8(R7), R0 | |
| | | | 59 | 50 | D1 00D38 | CMPL | R0, WIDTH | |
| | | | | 22 1B 00D3B | BLEQU | 130\$ | | |
| | | | | 5B D5 00D3D | TSTL | TERM_LENGTH | | |
| | | | | OF 15 00D3F | BLEQ | 129\$ | | |
| | 0A5C CE46 | 08 | BE | 14 | 5B 28 00D41 | MOV C3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 50 | BC | 3C 00D49 | MOVZWL | @TERM_DESC, R0 | |

| | | | | | | | | | |
|----|---------------------|----------|----------|----------|----------------|--------|----------------|---|------|
| SA | 20 | 56 6E | 50 00 | CO 2C | 00D4D 00D50 | 129\$: | ADDL2 MOVCS | RO, SIZE #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | 0A5C | CE46 | 00D55 | | | | |
| | | 56 | 5A | CO | 00D59 | | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A | DO | 00D5C | | MOVL | INDENT, LINE SIZE | |
| | 0A5C CE46 00000000' | 57 | 08 | CO | 00D5F | 130\$: | ADDL2 | #8, LINE SIZE | |
| | | EF | 08 | 28 | 00D62 | | MOVCS | #8, P.AFA, BUFFER[SIZE] | |
| | 40 | 56 | 08 | CO | 00D6D | | ADDL2 | #8, SIZE | |
| | | 58 | 08 | E5 | 00D70 | | BBCC | #8, FLAGS, 133\$ | 1300 |
| | | 28 | 28 | AE | 00D74 | | BLBC | 40(SP), 132\$ | 1301 |
| | | 50 | 08 | A7 | 00D78 | | MOVAB | 8(R7), RO | |
| | | 59 | 50 | D1 | 00D7C | | CMPL | RO, WIDTH | |
| | | | 22 | 1B | 00D7F | | BLEQU | 132\$ | |
| | | | 5B | D5 | 00D81 | | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | 00D83 | | BLEQ | 131\$ | |
| | 0A5C CE46 08 | 8E | 5B | 28 | 00D85 | | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 50 | 14 | BC | 3C | 00D8D | MOVZWL | @TERM_DESC, RO | |
| | | 56 | 50 | CO | 00D91 | | ADDL2 | RO, SIZE | |
| SA | 20 | 6E | 00 | 2C | 00D94 | 131\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | 0A5C | CE46 | 00D99 | | | | |
| | | 56 | 5A | CO | 00D9D | | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A | DO | 00DA0 | | MOVL | INDENT, LINE SIZE | |
| | | 57 | 08 | CO | 00DA3 | 132\$: | ADDL2 | #8, LINE SIZE | |
| | 0A5C CE46 00000000' | EF | 08 | 28 | 00DA6 | | MOVCS | #8, P.AFB, BUFFER[SIZE] | |
| | | 56 | 08 | CO | 00DB1 | | ADDL2 | #8, SIZE | |
| | 40 | 58 | 0A | E5 | 00DB4 | 133\$: | BBCC | #10, FLAGS, 136\$ | 1302 |
| | | 28 | 28 | AE | 00DB8 | | BLBC | 40(SP), 135\$ | 1303 |
| | | 50 | 07 | A7 | 00DBC | | MOVAB | 7(R7), RO | |
| | | 59 | 50 | D1 | 00DC0 | | CMPL | RO, WIDTH | |
| | | | 22 | 1B | 00DC3 | | BLEQU | 135\$ | |
| | | | 5B | D5 | 00DC5 | | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | 00DC7 | | BLEQ | 134\$ | |
| | 0A5C CE46 08 | 8E | 5B | 28 | 00DC9 | | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 50 | 14 | BC | 3C | 00DD1 | MOVZWL | @TERM_DESC, RO | |
| | | 56 | 50 | CO | 00DD5 | | ADDL2 | RO, SIZE | |
| SA | 20 | 6E | 00 | 2C | 00DD8 | 134\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | 0A5C | CE46 | 00DDD | | | | |
| | | 56 | 5A | CO | 00DE1 | | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A | DO | 00DE4 | | MOVL | INDENT, LINE SIZE | |
| | | 57 | 07 | CO | 00DE7 | 135\$: | ADDL2 | #7, LINE SIZE | |
| | 0A5C CE46 00000000' | EF | 07 | 28 | 00DEA | | MOVCS | #7, P.AFC, BUFFER[SIZE] | |
| | | 56 | 07 | CO | 00DF5 | | ADDL2 | #7, SIZE | |
| | 40 | 58 | 09 | E5 | 00DF8 | 136\$: | BBCC | #9, FLAGS, 139\$ | 1304 |
| | | 28 | 28 | AE | 00DFC | | BLBC | 40(SP), 138\$ | 1305 |
| | | 50 | 0A | A7 | 00E00 | | MOVAB | 10(R7), RO | |
| | | 59 | 50 | D1 | 00E04 | | CMPL | RO, WIDTH | |
| | | | 22 | 1B | 00E07 | | BLEQU | 138\$ | |
| | | | 5B | D5 | 00E09 | | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | 00E0B | | BLEQ | 137\$ | |
| | 0A5C CE46 08 | 8E | 5B | 28 | 00E0D | | MOVCS | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 50 | 14 | BC | 3C | 00E15 | MOVZWL | @TERM_DESC, RO | |
| | | 56 | 50 | CO | 00E19 | | ADDL2 | RO, SIZE | |
| SA | 20 | 6E | 00 | 2C | 00E1C | 137\$: | MOVCS | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | | 0A5C | CE46 | 00E21 | | | | |
| | | 56 | 5A | CO | 00E25 | | ADDL2 | INDENT, SIZE | |
| | | 57 | 5A | DO | 00E28 | | MOVL | INDENT, LINE SIZE | |
| | | 57 | 0A | CO | 00E2B | 138\$: | ADDL2 | #10, LINE_SIZE | |

| | | | | | | | |
|---------------------|----|-----------|----|-------|--------|--|------|
| 0A5C CE46 00000000' | EF | 0A | 28 | 00E2E | MOV C3 | #10, P.AFD, BUFFER[SIZE] | |
| 40 | 56 | 0A | C0 | 00E39 | ADD L2 | #10, SIZE | |
| | 58 | 0B | E5 | 00E3C | BBCC | #11, FLAGS, 142\$ | 1306 |
| | 28 | AE | E9 | 00E40 | BLBC | 40(SP), 141\$ | 1307 |
| | 50 | A7 | 9E | 00E44 | MOVAB | 12(R7), R0 | |
| | 59 | 50 | D1 | 00E48 | CMPL | R0, WIDTH | |
| | | 22 | 1B | 00E4B | BLEQU | 141\$ | |
| | | 5B | D5 | 00E4D | TSTL | TERM_LENGTH | |
| | | 0F | 15 | 00E4F | BLEQ | 140\$ | |
| 0A5C CE46 08 | BE | 5B | 28 | 00E51 | MOV C3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | 50 | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | 56 | | 50 | C0 | ADD L2 | R0, SIZE | |
| SA 20 | 6E | | 00 | 2C | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 0A5C CE46 | | 00E65 | | | |
| | 56 | SA | C0 | 00E69 | ADD L2 | INDENT, SIZE | |
| | 57 | SA | D0 | 00E6C | MOVL | INDENT, LINE SIZE | |
| 0A5C CE46 00000000' | 57 | OC | C0 | 00E6F | ADD L2 | #12, LINE SIZE | |
| | EF | OC | 28 | 00E72 | MOV C3 | #12, P.AFD, BUFFER[SIZE] | |
| | 56 | OC | C0 | 00E7D | ADD L2 | #12, SIZE | |
| | | 58 | B5 | 00E80 | TSTW | FLAGS | 1308 |
| | | 5C | 13 | 00E82 | BEQL | 145\$ | |
| | 28 | 28 | AE | E9 | BLBC | 40(SP), 144\$ | 1311 |
| | 50 | 07 | A7 | 9E | MOVAB | 7(R7), R0 | |
| | 59 | | 50 | D1 | CMPL | R0, WIDTH | |
| | | | 22 | 1B | BLEQU | 144\$ | |
| | | | 5B | D5 | TSTL | TERM_LENGTH | |
| | | | 0F | 15 | BLEQ | 143\$ | |
| 0A5C CE46 08 | BE | 5B | 28 | 00E95 | MOV C3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | 50 | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | 56 | | 50 | C0 | ADD L2 | R0, SIZE | |
| SA 20 | 6E | | 00 | 2C | MOV C5 | #0, (SP), #32, INDENT, BUFFER[SIZE] | |
| | | 0A5C CE46 | | 00EA9 | | | |
| | 56 | SA | C0 | 00EAD | ADD L2 | INDENT, SIZE | |
| | 57 | SA | D0 | 00EB0 | MOVL | INDENT, LINE SIZE | |
| | 57 | 07 | C0 | 00EB3 | ADD L2 | #7, LINE SIZE | |
| FEF8 CD | | 07 | B0 | 00EB6 | MOVW | #7, FAO_DESCR | 1312 |
| FEFC CD | | 0A5C CE46 | 9E | 00EBB | MOVAB | BUFFER[SIZE], FAO_DESCR+4 | 1313 |
| 7E | | 58 | 3C | 00EC3 | MOVZWL | FLAGS, -(SP) | 1317 |
| | | FEF8 CD | 9F | 00EC6 | PUSHAB | FAO_DESCR | |
| | | FEF8 CD | 9F | 00ECA | PUSHAB | FAO_DESCR | |
| | | FEF8 EF | 9F | 00ECE | PUSHAB | P.AFD | |
| 00000000G 00 | | 04 | FB | 00ED4 | CALLS | #4, SYSSFAO | |
| | 56 | 07 | C0 | 00EDB | ADD L2 | #7, SIZE | 1318 |
| | | 06 | 11 | 00EDE | BRB | 146\$ | 1308 |
| 0A5B CE46 | | 2C | 90 | 00EE0 | MOV B | #44, BUFFER-1[SIZE] | 1320 |
| 03 | | 20 | AE | E8 | BLBS | ACCESS_MASK, 147\$ | 1325 |
| | | 01D6 | 31 | 00EEA | BRW | 171\$ | |
| | | FF04 CD | D5 | 00EED | TSTL | LOCAL_ACE+4 | 1328 |
| | | 12 | 12 | 00EF1 | BNEQ | 150\$ | |
| | 03 | 24 | AE | E8 | BLBS | AUDIT_MASK, 149\$ | 1329 |
| | | 0189 | 31 | 00EF7 | BRW | 168\$ | |
| | | FF02 CD | E8 | 00EFA | BLBS | LOCAL_ACE+2, 150\$ | 1330 |
| F2 FF02 | | 01 | E1 | 00EFF | BBC | #1, LOCAL_ACE+2, 148\$ | 1331 |
| | | 28 | AE | E9 | BLBC | 40(SP), 152\$ | 1334 |
| | | 07 | A7 | 9E | MOVAB | 7(R7), R0 | |
| | | | 50 | D1 | CMPL | R0, WIDTH | |
| | | | 22 | 1B | BLEQU | 152\$ | |

| | | | | | | | | | |
|------|------|-----------|----|-----------|-----------|-------|--------|--|--------------------|
| | | | | 5B | D5 | 00F12 | TSTL | TERM_LENGTH | |
| | | | | 0F | 15 | 00F14 | BLEQ | 151\$ | |
| 0A5C | CE46 | 08 | BE | 5B | 28 | 00F16 | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 50 | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| 5A | 20 | | 56 | 50 | C0 | 00F22 | ADDL2 | R0, SIZE | |
| | | | 6E | 00 | 2C | 00F25 | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | 151\$: |
| | | | | 0A5C | CE46 | 00F2A | | | |
| | | | 56 | 5A | C0 | 00F2E | ADDL2 | INDENT, SIZE | |
| | | | 57 | 5A | D0 | 00F31 | MOVL | INDENT, LINE_SIZE | |
| | | | 57 | 07 | C0 | 00F34 | ADDL2 | #7, LINE_SIZE | 152\$: |
| 0A5C | CE46 | 00000000' | EF | 07 | 28 | 00F37 | MOV3 | #7, P.AFR, BUFFER[SIZE] | |
| | | | 56 | 07 | C0 | 00F42 | ADDL2 | #7, SIZE | |
| | | | | 58 | D4 | 00F45 | CLRL | J | |
| | 03 | FF04 | CD | 58 | E0 | 00F47 | BBS | J, LOCAL_ACE+4, 154\$ | 1335 |
| | | | | 0099 | 31 | 00F4D | BRW | 160\$ | 1338 |
| | | | 50 | 00000000' | EF | D0 | 00F50 | MOVL | BIT_NAME_TABLE, R0 |
| | | | | 28 | 13 | 00F57 | BEQL | 156\$ | 1341 |
| | | | | 6048 | 7F | 00F59 | PUSHAQ | (R0)[J] | 1344 |
| | 9E | | 08 | 00 | 0C | 00F5C | PROBER | #0, #8, @ (SP)+ | |
| | | | | 1C | 13 | 00F60 | BEQL | 155\$ | |
| | | | | 6048 | 7E | 00F62 | MOVAQ | (R0)[J], BIT_NAME_DESC | 1345 |
| | 54 | 1C | AE | 04 | C1 | 00F67 | ADDL3 | #4, BIT_NAME_DESC, R4 | 1347 |
| | | 1C | AE | 64 | D0 | 00F6C | MOVL | (R4), R0 | |
| | | | 50 | 1C | BE | 3C | MOVZWL | @BIT_NAME_DESC, R1 | |
| | | | 51 | 53 | D4 | 00F73 | CLRL | R3 | |
| | | | | 00000000G | 00 | 16 | 00F75 | JSB | EXESPROBER |
| | | | 0C | 50 | E8 | 00F7B | BLBS | R0, 157\$ | |
| | | | | 0183 | 31 | 00F7E | BRW | 172\$ | 1349 |
| | | | 1C | AE | 00000000' | EF48 | D0 | 00F81 | 156\$: |
| | | | 30 | 28 | AE | E9 | 00F8A | 157\$: | 1351 |
| | | | 50 | 1C | BE | 3C | 00F8E | BLBC | 40(SP), 159\$ |
| | | | 50 | 01 | A047 | 9E | 00F92 | MOVZWL | @BIT_NAME_DESC, R0 |
| | | | 59 | 50 | D1 | 00F97 | MOVAB | 1(R0)[LINE_SIZE], R0 | |
| | | | | 22 | 1B | 00F9A | CPL | R0, WIDTH | |
| | | | | 5B | D5 | 00F9C | BLEQU | 159\$ | |
| | | | | 0F | 15 | 00F9E | TSTL | TERM_LENGTH | |
| 0A5C | CE46 | 08 | BE | 5B | 28 | 00FA0 | BLEQ | 158\$ | |
| | | | 50 | 14 | BC | 3C | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | | 56 | 50 | C0 | 00FAC | MOVZWL | @TERM_DESC, R0 | |
| 5A | 20 | | 6E | 00 | 2C | 00FAF | ADDL2 | R0, SIZE | |
| | | | | 0A5C | CE46 | 00FB4 | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | 158\$: |
| | | | 56 | 5A | C0 | 00FB8 | | | |
| | | | 57 | 5A | D0 | 00FBB | ADDL2 | INDENT, SIZE | |
| | | | | 1C | BE | 3C | MOVL | INDENT, LINE_SIZE | |
| | | | 20 | 01 | C1 | 00FC3 | MOVZWL | @BIT_NAME_DESC, 32(SP) | 159\$: |
| | 50 | 20 | AE | 50 | C0 | 00FC8 | ADDL3 | #1, 32(SPT, R0 | |
| | | | 57 | 04 | C1 | 00FCB | ADDL2 | R0, LINE_SIZE | |
| | 7E | 1C | AE | 9E | DD | 00FDD | ADDL3 | #4, BIT_NAME_DESC, -(SP) | 1354 |
| | | | | 24 | AE | 28 | PUSHL | @(SP)+ | |
| 0A5C | CE46 | | 9E | 24 | AE | 28 | MOV3 | 36(SP), @ (SP)+, BUFFER[SIZE] | |
| | 50 | | 56 | 20 | AE | C1 | ADDL3 | 32(SP), SIZE, R0 | 1355 |
| | | | | 0A5C | CE40 | 28 | MOVB | #43, BUFFER[R0] | |
| | | | 56 | 01 | A0 | 9E | MOVAB | 1(R0), SIZE | 1356 |
| FF5B | 58 | | 01 | 1F | F1 | 00FE9 | | #31, #1, J, 153\$ | 1335 |
| | | | 03 | 24 | AE | E8 | ACBL | AUDIT_MASK, 162\$ | 1359 |
| | | | | 00CD | 31 | 00FF3 | BLBS | 171\$ | |
| | | | 40 | FF02 | CD | E9 | BRW | LOCAL_ACE+2, 165\$ | 1362 |
| | | | 28 | 28 | AE | E9 | BLBC | 40(SPT, 164\$ | |

| | | | | | | | | | |
|----|---------------------|----|----|-----------|----|-------|--------|--|--------|
| | | 50 | 08 | A7 | 9E | 00FFF | MOVAB | 8(R7), R0 | |
| | | 59 | | 50 | D1 | 01003 | CMPL | R0, WIDTH | |
| | | | | 22 | 1B | 01006 | BLEQU | 164\$ | |
| | | | | 5B | D5 | 01008 | TSTL | TERM_LENGTH | |
| | | | | OF | 15 | 0100A | BLEQ | 163\$ | |
| | 0A5C CE46 | 08 | BE | 5B | 28 | 0100C | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 50 | | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | | 56 | | 50 | C0 | 01018 | ADDL2 | R0, SIZE | |
| SA | 20 | 6E | | 00 | 2C | 0101B | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | 163\$: |
| | | | | 0A5C CE46 | | 01020 | | | |
| | | 56 | | 5A | C0 | 01024 | ADDL2 | INDENT, SIZE | |
| | | 57 | | 5A | D0 | 01027 | MOVL | INDENT, LINE_SIZE | |
| | 0A5C CE46 00000000' | 57 | | 08 | C0 | 0102A | ADDL2 | #8, LINE_SIZE | 164\$: |
| | | EF | | 08 | 28 | 0102D | MOV3 | #8, P.AFT, BUFFER[SIZE] | |
| | | 56 | | 08 | C0 | 01038 | ADDL2 | #8, SIZE | |
| | B2 FF02 | CD | | 01 | E1 | 0103B | BBC | #1, LOCAL ACE+2, 161\$ | 165\$: |
| | | 2B | | 28 | AE | E9 | BLBC | 40(SP), 167\$ | 1363 |
| | | 50 | | 08 | A7 | 9E | MOVAB | 8(R7), R0 | |
| | | 59 | | 50 | D1 | 01049 | CMPL | R0, WIDTH | |
| | | | | 22 | 1B | 0104C | BLEQU | 167\$ | |
| | | | | 5B | D5 | 0104E | TSTL | TERM_LENGTH | |
| | | | | OF | 15 | 01050 | BLEQ | 166\$ | |
| | 0A5C CE46 | 08 | BE | 5B | 28 | 01052 | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 50 | | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | | 56 | | 50 | C0 | 0105E | ADDL2 | R0, SIZE | |
| SA | 20 | 6E | | 00 | 2C | 01061 | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | 166\$: |
| | | | | 0A5C CE46 | | 01066 | | | |
| | | 56 | | 5A | C0 | 0106A | ADDL2 | INDENT, SIZE | |
| | | 57 | | 5A | D0 | 0106D | MOVL | INDENT, LINE_SIZE | |
| | 0A5C CE46 00000000' | 57 | | 08 | C0 | 01070 | ADDL2 | #8, LINE_SIZE | 167\$: |
| | | EF | | 08 | 28 | 01073 | MOV3 | #8, P.AFT, BUFFER[SIZE] | |
| | | 56 | | 08 | C0 | 0107E | ADDL2 | #8, SIZE | |
| | | | | 40 | 11 | 01081 | BRB | 171\$ | 1328 |
| | | 2B | | 28 | AE | E9 | BLBC | 40(SP), 170\$ | 1366 |
| | | 50 | | 0C | A7 | 9E | MOVAB | 12(R7), R0 | |
| | | 59 | | 50 | D1 | 0108B | CMPL | R0, WIDTH | |
| | | | | 22 | 1B | 0108E | BLEQU | 170\$ | |
| | | | | 5B | D5 | 01090 | TSTL | TERM_LENGTH | |
| | | | | OF | 15 | 01092 | BLEQ | 169\$ | |
| | 0A5C CE46 | 08 | BE | 5B | 28 | 01094 | MOV3 | TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] | |
| | | 50 | | 14 | BC | 3C | MOVZWL | @TERM_DESC, R0 | |
| | | 56 | | 50 | C0 | 010A0 | ADDL2 | R0, SIZE | |
| SA | 20 | 6E | | 00 | 2C | 010A3 | MOV3 | #0, (SP), #32, INDENT, BUFFER[SIZE] | 169\$: |
| | | | | 0A5C CE46 | | 010A8 | | | |
| | | 56 | | 5A | C0 | 010AC | ADDL2 | INDENT, SIZE | |
| | | 57 | | 5A | D0 | 010AF | MOVL | INDENT, LINE_SIZE | |
| | 0A5C CE46 00000000' | 57 | | 0C | C0 | 010B2 | ADDL2 | #12, LINE_SIZE | 170\$: |
| | | EF | | 0C | 28 | 010B5 | MOV3 | #12, P.AFT, BUFFER[SIZE] | |
| | | 56 | | 0C | C0 | 010C0 | ADDL2 | #12, SIZE | |
| | 0A5B CE46 | | | 29 | 90 | 010C3 | MOVB | #41, BUFFER-1[SIZE] | 171\$: |
| | 0C BC | 08 | | 00 | 0C | 010C9 | PROBER | #0, #8, @ACL_STRING | 1371 |
| | | | | 34 | 13 | 010CE | BEQL | 172\$ | 1375 |
| | 54 0C | AC | | 04 | C1 | 010D0 | ADDL3 | #4, ACL_STRING, R4 | 1378 |
| | | 50 | | 64 | D0 | 010D5 | MOVL | (R4), R0 | |
| | | 51 | | 6E | D0 | 010D8 | MOVL | ACL_STRING_LEN, R1 | |
| | | | | 53 | D4 | 010DB | CLRL | R3 | |
| | 00000000G | | | 00 | 16 | 010DD | JSB | EXESPROBEW | |

| | | | | | | | | | | |
|----|----|------|----|------|----|----------|--------|---|---|------|
| 6E | 57 | 0C | 1E | 50 | E9 | 010E3 | BLBC | R0, 172\$ | : | |
| | 00 | 0ASC | AC | 04 | C1 | 010E6 | ADDL3 | #4, ACL_STRING, R7 | : | 1379 |
| | | | CE | 56 | 2C | 010EB | MOVCS | SIZE, BUFFER, #0, ACL_STRING_LEN, @R7)+ | : | |
| | | | | 97 | | 010F2 | | | : | |
| | | | 50 | 08 | AC | D0 010F3 | MOVL | ACL_LENGTH, R0 | : | 1383 |
| | | | | | 0F | 13 010F7 | BEQL | 173\$ | : | |
| | 60 | | 04 | | 00 | 0D 010F9 | PROBEW | #0, #4, (R0) | : | 1384 |
| | | | | | 05 | 13 010FD | BEQL | 172\$ | : | |
| | | | 60 | | 56 | D0 010FF | MOVL | SIZE, (R0) | : | 1385 |
| | | | | | 04 | 11 01102 | BRB | 173\$ | : | |
| | | | 50 | | 0C | D0 01104 | MOVL | #12, R0 | : | |
| | | | | | 04 | 01107 | RET | | : | |
| | | | 6E | | 56 | D1 01108 | CMPL | SIZE, ACL_STRING_LEN | : | 1386 |
| | | | | | 06 | 15 0110B | BLEQ | 175\$ | : | |
| | | | 50 | 0601 | 8F | 3C 0110D | MOVZWL | #1537, R0 | : | 1387 |
| | | | | | 04 | 01112 | RET | | : | |
| | | | 50 | | 01 | D0 01113 | MOVL | #1, R0 | : | |
| | | | | | 04 | 01116 | RET | | : | 1389 |

; Routine Size: 4375 bytes, Routine Base: \$CODE\$ + 0201


```
1395 1390 1 %SBTTL ' $CHANGE_ACL system service'
1396 1391 1 GLOBAL ROUTINE SYSS$CHANGE_ACL (CHANNEL, OBJECT TYPE, OBJECT NAME,
1397 1392 1                                     ITEM_LIST, ACCESS_MODE, RESERVED, CONTEXT) =
1398 1393 1
1399 1394 1 ++
1400 1395 1
1401 1396 1 FUNCTIONAL DESCRIPTION:
1402 1397 1
1403 1398 1     This routine changes (or reads) the ACL associated with any of the
1404 1399 1     defined objects within the system.
1405 1400 1
1406 1401 1     Note:   Since the length of the ACE is part of the data returned to
1407 1402 1           the caller, the return length parameter of the item list
1408 1403 1           is unused.
1409 1404 1
1410 1405 1     There are basically two types of objects that can have their ACL
1411 1406 1     twiddled. There are those objects that require an agent to do the work
1412 1407 1     (e.g. files by the XQP) and the others where it is simply necessary to
1413 1408 1     locate the ACL queue segment list head. In the latter case, the work
1414 1409 1     is done here.
1415 1410 1
1416 1411 1 CALLING SEQUENCE:
1417 1412 1     SYSS$CHANGE_ACL (ARG1, ARG2, ARG3, ARG4, ARG5, ARG6, ARG7)
1418 1413 1
1419 1414 1 INPUT PARAMETERS:
1420 1415 1     ARG1: number of the channel assigned to the object or 0 if
1421 1416 1           the object is specified by ARG2 and ARG3
1422 1417 1     ARG2: address of an object type code
1423 1418 1     ARG3: address of an object name descriptor
1424 1419 1     ARG4: address of a list of item descriptors
1425 1420 1     ARG5: address of an access mode longword (used to validate the
1426 1421 1           item list and I/O status block)
1427 1422 1     ARG6: reserved for future use
1428 1423 1     ARG7: address of a context longword
1429 1424 1
1430 1425 1 IMPLICIT INPUTS:
1431 1426 1     none
1432 1427 1
1433 1428 1 OUTPUT PARAMETERS:
1434 1429 1     ARG7: address of a context longword
1435 1430 1
1436 1431 1 IMPLICIT OUTPUTS:
1437 1432 1     none
1438 1433 1
1439 1434 1 ROUTINE VALUE:
1440 1435 1     SS$NORMAL - if the requested action completed successfully
1441 1436 1     SS$NOPRIV - if the requestor did not have privileges for the
1442 1437 1                  requested action
1443 1438 1
1444 1439 1 SIDE EFFECTS:
1445 1440 1     The context longword is modified as necessary based upon the
1446 1441 1     action requested.
1447 1442 1
1448 1443 1 --
1449 1444 1
1450 1445 2 BEGIN
1451 1446 2
```

```
1452 1447 2 MAP
1453 1448      CHANNEL      : WORD,
1454 1449      OBJECT_NAME : REF $BBLOCK,
1455 1450      ITEM_LIST    : REF BLOCKVECTOR [, ITMSS_ITEM, BYTE];
1456 1451
1457 1452 LOCAL
1458 1453      STATUS          ! Local routine return status
1459 1454      STATUS2       ! Temp status, may overwrite STATUS
1460 1455      PSL           : $BBLOCK [4],      ! Local copy of PSL
1461 1456      LOCAL_OBJTYP, ! Local copy of object type code
1462 1457      LOCAL_IOSB    : VECTOR [4, WORD], ! Local copy of the I/O status block
1463 1458      LOCAL_LOCKID,  ! Local copy of the lock-id
1464 1459      OBJECT_DESC    : VECTOR [2],      ! Descriptor of object name
1465 1460      SHARE         : BYTE,           ! Whether to allow sharing or not
1466 1461      ITEM_COUNT,  ! Number of items in the list
1467 1462      ITEM_CODE,    ! Code from item list entry
1468 1463      ITEM_SIZE,   ! Size from item list entry
1469 1464      ITEM_ADDR,   ! Buffer addr from item list entry
1470 1465      LOCAL_CHANNEL : WORD,         ! Local copy of user's channel
1471 1466      IO_CHANNEL   : WORD,         ! Object's channel
1472 1467      FILE_FAB      : $FAB_DECL,    ! Object file's FAB
1473 1468      FILE_NAME     : $NAM_DECL,    ! Object file's NAME block
1474 1469      FILE_EXP_NAME  : $BBLOCK [NAMSC_MAXRSS], ! Expanded name storage
1475 1470      FILE_RES_NAME  : $BBLOCK [NAMSC_MAXRSS], ! Resultant name storage
1476 1471      FILE_FIB_DESC  : $BBLOCK [DSCSC_S_BLN], ! File FIB descriptor
1477 1472      FILE_FIB       : $BBLOCK [FIBSC_LENGTH], ! File FIB storage
1478 1473      DVI_ATR_LIST   : BLOCKVECTOR [2, ITMSS_ITEM, BYTE], ! $GETDVI item list
1479 1474      ACP_ATR_PTR    ! Pointer into ACP attribute list
1480 1475      ACP_ATR_LIST  : REF BLOCKVECTOR [, 8, BYTE], ! ACP attribute list
1481 1476      ACL_TO_ATR_TAB : VECTOR [MAX_ACL_ATR + 1] ! ATRSC to ACLSC xlate
1482 1477      INITIAL (0,
1483 1478          ATRSC_ADDACLENT,
1484 1479          ATRSC_DELEACLENT,
1485 1480          ATRSC_MODACLENT,
1486 1481          ATRSC_FNDACLENT,
1487 1482          ATRSC_FNDACETYP,
1488 1483          ATRSC_DELETEACL,
1489 1484          ATRSC_READACL,
1490 1485          ATRSC_ACLLENGTH,
1491 1486          ATRSC_READACE),
1492 1487      FUNCTION_CODE, ! QIOW function code
1493 1488      CMK_ARG_LIST    : VECTOR [5], ! Also, ACL dispatch code
1494 1489      ! $CMKRNL arg list
1495 1490
1496 1491      ! See if an access mode parameter was given.
1497 1492
1498 1493      CHANGE_ACMODE = 0;
1499 1494      IF .ACCESS_MODE NEQA 0
1500 1495      THEN IF PROBER (%REF (0), %REF (4), .ACCESS_MODE)
1501 1496      THEN CHANGE_ACMODE = .ACCESS_MODE
1502 1497      ELSE RETURN %SS$_ACCVIO;
1503 1498
1504 1499      MOVPSL (PSL);
1505 1500      CALL_ACMODE = .PSL[PSL$V_PRVMOD];
1506 1501      CHANGE_ACMODE = MAXU (.CHANGE_ACMODE, .CALL_ACMODE);
1507 1502
1508 1503      ! Determine the validity of the access mode parameter.
```

```
1509 1504 2
1510 1505 2 IF .CHANGE_ACMODE GTRU PSL$C_USER THEN RETURN SS$_BADPARAM;
1511 1506 2
1512 1507 2 ! Get the supplied channel, if any, and verify it.
1513 1508 2
1514 1509 2 IO_CHANNEL = LOCAL_CHANNEL = .CHANNEL;
1515 1510 2 IF .IO_CHANNEL NEQ 0
1516 1511 2 THEN
1517 1512 2 BEGIN
1518 1513 2 STATUS = IOC$VERIFYCHAN (.IO_CHANNEL);
1519 1514 2 IF NOT .STATUS THEN RETURN .STATUS;
1520 1515 2 END;
1521 1516 2
1522 1517 2 ! Get the object type code.
1523 1518 2
1524 1519 2 IF .OBJECT_TYPE NEQA 0
1525 1520 2 THEN (IF PROBER (%REF (0), %REF (4), .OBJECT_TYPE)
1526 1521 2 THEN LOCAL OBJTYP = .OBJECT_TYPE
1527 1522 2 ELSE RETURN SS$_ACCVIO)
1528 1523 2 ELSE RETURN SS$_INSFARG;
1529 1524 2
1530 1525 2 ! Check the validity of the object type code.
1531 1526 2
1532 1527 2 IF .LOCAL_OBJTYP LSSU MIN_OBJECT_TYPE
1533 1528 2 OR .LOCAL_OBJTYP GTRU MAX_OBJECT_TYPE
1534 1529 2 THEN RETURN SS$_BADPARAM;
1535 1530 2
1536 1531 2 ! Probe the object name if supplied.
1537 1532 2
1538 1533 2 IF .OBJECT_NAME NEQA 0
1539 1534 2 THEN
1540 1535 2 BEGIN
1541 1536 2 IF NOT PROBER (%REF (0), %REF (DSC$C_S_BLN), .OBJECT_NAME)
1542 1537 2 THEN RETURN SS$_ACCVIO;
1543 1538 2 OBJECT_DESC[0] = .OBJECT_NAME[DSC$W_LENGTH];
1544 1539 2 OBJECT_DESC[1] = .OBJECT_NAME[DSC$A_POINTER];
1545 1540 2 IF NOT EXE$PROBER (0, .OBJECT_DESC[0], .OBJECT_DESC[1])
1546 1541 2 THEN RETURN SS$_ACCVIO;
1547 1542 2 END
1548 1543 2 ELSE
1549 1544 2 BEGIN
1550 1545 2 OBJECT_DESC[0] = 0;
1551 1546 2 OBJECT_DESC[1] = 0;
1552 1547 2 END;
1553 1548 2
1554 1549 2 ! Get any value supplied for the context parameter.
1555 1550 2
1556 1551 2 ACL_CONTEXT = 0;
1557 1552 2 IF .CONTEXT NEQA 0
1558 1553 2 THEN IF PROBER (%REF (0), %REF (4), .CONTEXT)
1559 1554 2 THEN ACL_CONTEXT = .CONTEXT
1560 1555 2 ELSE RETURN SS$_ACCVIO;
1561 1556 2
1562 1557 2 ! Count the number of items in the item list.
1563 1558 2
1564 1559 2 SHARE = 1;
1565 1560 2 INCR J FROM 0
```

! Assume shared access

```
1566 1561 2 DO IF PROBER (XREF (0), XREF (ITM$$ ITEM), ITEM_LIST[J, 0,0,0,0])
1567 1562 3 THEN (IF .ITEM_LIST[J, ITMSW_BUFSIZ] EQL 0
1568 1563 4 THEN
1569 1564 5 BEGIN
1570 1565 6 ITEM COUNT = .J;
1571 1566 7 EXITCOOP;
1572 1567 8 END
1573 1568 9 ELSE
1574 1569 10 BEGIN
1575 1570 11 IF .ITEM_LIST[J, ITMSW_ITMCD] EQL ACL$C_ADDACLENT
1576 1571 12 OR .ITEM_LIST[J, ITMSW_ITMCD] EQL ACL$C_DELACLENT
1577 1572 13 OR .ITEM_LIST[J, ITMSW_ITMCD] EQL ACL$C_MODACLENT
1578 1573 14 OR .ITEM_LIST[J, ITMSW_ITMCD] EQL ACL$C_DELETEACL
1579 1574 15 THEN SHARE = 0;
1580 1575 16 END)
1581 1576 17 ELSE RETURN SS$_ACCVIO;
1582 1577 18
1583 1578 19 ! Initialize all common (to both types of objects) storage.
1584 1579 20
1585 1580 21 CH$FILL (0, 2*ITM$$ ITEM, DVI_ATR_LIST);
1586 1581 22 CH$FILL (0, DSC$C_S_BLN, LOCK_RESNAM);
1587 1582 23 LOCAL_LOCKID = 0;
1588 1583 24
1589 1584 25 ! Set up the lock resource name prefix.
1590 1585 26
1591 1586 27 LOCK_RESNAM[DSC$W_LENGTH] = RSN_S_PREFIX;
1592 1587 28 LOCK_RESNAM[DSC$A_POINTER] = RESNAM_TEXT;
1593 1588 29 CH$COPY (.SBBLOCK[.LOCK_PREFIX[LOCAL_OBJTYP], DSC$W_LENGTH],
1594 1589 30 .SBBLOCK[.LOCK_PREFIX[LOCAL_OBJTYP], DSC$A_POINTER],
1595 1590 31 0,
1596 1591 32 RSN_S_PREFIX, RESNAM_TEXT);
1597 1592 33
1598 1593 34 ! If the call is from user mode, take out a lock to form the parent lock ID
1599 1594 35 ! for all ACL locks. This facilitates releasing them at image rundown.
1600 1595 36
1601 1596 37 IF .CALL_ACMODE EQL PSL$C_USER
1602 1597 38 THEN
1603 1598 39 IF .PARENT_ID EQL 0
1604 1599 40 THEN
1605 1600 41 BEGIN
1606 1601 42 STATUS = $CMKRNL (ROUTIN = GET_PARENT_LOCK);
1607 1602 43 IF NOT .STATUS THEN RETURN .STATUS;
1608 1603 44 END;
1609 1604 45
1610 1605 46 ! Do any initial setup for the object. For files, this means opening the
1611 1606 47 ! specified file if it is not already open. For devices, this means assigning
1612 1607 48 ! a channel is one is not already assigned. For most other objects, nothing
1613 1608 49 ! special is needed.
1614 1609 50
1615 1610 51 CASE .LOCAL_OBJTYP FROM MIN_OBJECT_TYPE TO MAX_OBJECT_TYPE OF
1616 1611 52 SET
1617 1612 53 [ACL$C_FILE]:
1618 1613 54 BEGIN
1619 1614 55
1620 1615 56 ! Initialize storage.
1621 1616 57
1622 1617 58 CH$FILL (0, FIB$C_LENGTH, FILE_FIB);
```



```
1623      CH$FILL (0, DSC$C_S_BLN, FILE_FIB_DESC);
1624      FILE_FIB_DESC[DSC$W_LENGTH] = FIB$C_LENGTH;
1625      FILE_FIB_DESC[DSC$A_POINTER] = FILE_FIB;
1626      FILE_FIB[FIB$B_AGENT_MODE] = .CHANGE_ACMODE;
1627
1628      ! If the file is not accessed, do it now.
1629
1630      IF .IO_CHANNEL EQL 0
1631      THEN
1632      BEGIN
1633      $FAB_INIT (FAB = FILE_FAB,
1634                FNS = .OBJECT_DESC[0],
1635                FNA = .OBJECT_DESC[1],
1636                FOP = UFO,
1637                NAM = FILE_NAM);
1638      $NAM_INIT (NAM = FILE_NAM,
1639                ESA = FILE_EXP_NAME,
1640                ESS = NAM$C_MAXRSS,
1641                RSA = FILE_RES_NAME,
1642                RSS = NAM$C_MAXRSS);
1643
1644      IF .SHARE
1645      THEN
1646      BEGIN
1647      FILE_FAB[FAB$B_SHR] = FAB$M_GET OR FAB$M_PUT OR FAB$M_UPI;
1648      FILE_FAB[FAB$B_FAC] = FAB$M_GET;
1649      END
1650      ELSE
1651      BEGIN
1652      FILE_FAB[FAB$B_SHR] = FAB$M_NIL;
1653      FILE_FAB[FAB$B_FAC] = FAB$M_GET OR FAB$M_PUT;
1654      END;
1655      FILE_FAB[FAB$V_FILE_MODE] = .CHANGE_ACMODE;
1656
1657      STATUS = $OPEN (FAB = FILE_FAB);
1658      IO_CHANNEL = .FILE_FAB[FAB$L_STV];
1659      END
1660      ELSE STATUS = SS$NORMAL;
1661
1662      ! Now that a channel has been assigned to the file, do a simple access to
1663      ! fill the fib. This is needed to get the file-id used to build the lock name.
1664
1665      IF .STATUS
1666      THEN
1667      BEGIN
1668      STATUS = $QIOW (CHAN = .IO_CHANNEL,
1669                     FUNC = IOS_ACCESS,
1670                     IOSB = LOCAL_IOSB,
1671                     P1 = FILE_FIB_DESC);
1672
1673      IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
1674      END;
1675
1676      [ACL$C_DEVICE]:
1677      BEGIN
1678
1679      ! If necessary assign a channel to the specified device.
```

```
1680 1675 3      IF .IO_CHANNEL EQL 0
1681 1676 3      THEN
1682 1677 4          BEGIN
1683 1678 4              IF .OBJECT_DESC[0] EQL 0
1684 1679 4                  THEN RETURN SSS_INSFARG;
1685 1680 4                  STATUS = $ASSIGN (DEVNAM = OBJECT_DESC,
1686 1681 4                      CHAN = IO_CHANNEL);
1687 1682 4                  END;
1688 1683 4
1689 1684 4      ! Now that there is a channel to the device, locate the ACL queue head.
1690 1685 4
1691 1686 4      IF .STATUS
1692 1687 4      THEN
1693 1688 4          BEGIN
1694 1689 4              CMK_ARG_LIST[0] = 1;                ! Number of args
1695 1690 4              CMK_ARG_LIST[1] = .IO_CHANNEL;    ! Channel number
1696 1691 4              STATUS = $CMKRN (ROUTIN = GET_UCB_ACL,
1697 1692 4                  ARGST = CMK_ARG_LIST);
1698 1693 4          END;
1699 1694 4      END;
1700 1695 4
1701 1696 4      [ACL$C_JOBCTL_QUEUE]:
1702 1697 4          BEGIN
1703 1698 4              STATUS = SSS_BADPARAM;
1704 1699 4          END;
1705 1700 4
1706 1701 4      [ACL$C_COMMON_EF_CLUSTER]:
1707 1702 4          BEGIN
1708 1703 4              IF .OBJECT_DESC[0] EQL 0
1709 1704 4                  THEN RETURN SSS_INSFARG;
1710 1705 4              CMK_ARG_LIST[0] = 1;                ! Number of args
1711 1706 4              CMK_ARG_LIST[1] = OBJECT_DESC;    ! Cluster name descr
1712 1707 4              STATUS = $CMKRN (ROUTIN = GET_CEB_ACL,
1713 1708 4                  ARGST = CMK_ARG_LIST);
1714 1709 4          END;
1715 1710 4
1716 1711 4      [ACL$C_LOGICAL_NAME_TABLE]:
1717 1712 4          BEGIN
1718 1713 4              IF .OBJECT_DESC[0] EQL 0
1719 1714 4                  THEN RETURN SSS_INSFARG;
1720 1715 4              CMK_ARG_LIST[0] = 1;                ! Number of args
1721 1716 4              CMK_ARG_LIST[1] = OBJECT_DESC;    ! Logical name table descr
1722 1717 4              STATUS = $CMKRN (ROUTIN = GET_LMT_ACL,
1723 1718 4                  ARGST = CMK_ARG_LIST);
1724 1719 4          END;
1725 1720 4
1726 1721 4      [ACL$C_PROCESS]:
1727 1722 4          BEGIN
1728 1723 4              IF .OBJECT_DESC[0] EQL 0
1729 1724 4                  THEN RETURN SSS_INSFARG;
1730 1725 4              CMK_ARG_LIST[0] = 1;                ! Number of args
1731 1726 4              CMK_ARG_LIST[1] = OBJECT_DESC;    ! Process name descr
1732 1727 4              STATUS = $CMKRN (ROUTIN = GET_PCB_ACL,
1733 1728 4                  ARGST = CMK_ARG_LIST);
1734 1729 4          END;
1735 1730 4
1736 1731 2      [ACL$C_GLOBAL_SECTION]:
```

```
1737      BEGIN
1738      IF .OBJECT_DESC[0] EQL 0
1739      THEN RETURN $$$_INSFARG;
1740      CMK_ARG_LIST[0] = 1;
1741      CMK_ARG_LIST[1] = OBJECT_DESC;
1742      STATUS = $CMKRNL (ROUTIN = GET_GBL_ACL,
1743      ARGST = CMK_ARG_LIST);
1744      END;
1745
1746      [INRANGE, OTRANGE]:      STATUS = $$$_BADPARAM;
1747      TES;
1748      ! If any error have occurred, leave now.
1749
1750      IF NOT .STATUS
1751      THEN
1752      BEGIN
1753      IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1754      RETURN .STATUS;
1755      END;
1756      ! Now that the device has been identified, and a channel assigned if needed,
1757      ! form the remainder of the lock resource name. Then do the appropriate lock
1758      ! or unlock.
1759
1760      IF .LOCAL_OBJTYP EQL ACLSC_FILE OR .LOCAL_OBJTYP EQL ACLSC_DEVICE
1761      THEN
1762      BEGIN
1763      LOCAL      TMP_LEN;
1764      ! Build the remaining portion of the lock name.
1765
1766      DVI_ATR_LIST[0, ITMSW_ITMCD] = DVI$_DEVLOCKNAM;
1767      DVI_ATR_LIST[0, ITMSW_BUFSIZ] = 31 - RSN $ PREFIX;
1768      DVI_ATR_LIST[0, ITMSL_BUFADR] = RESNAM_TEXT[RSN_T_DEVNAM];
1769      DVI_ATR_LIST[0, ITMSL_RETLEN] = TMP_LEN;
1770      STATUS = $GETDVI (CHAN = .IO_CHANNEL,
1771      ITMLST = DVI_ATR_LIST,
1772      IOSB = LOCAL_IOSB);
1773      IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
1774      IF NOT .STATUS
1775      THEN
1776      BEGIN
1777      IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1778      RETURN .STATUS;
1779      END;
1780
1781      LOCK_RESNAM[DSCSW_LENGTH] = .LOCK_RESNAM[DSCSW_LENGTH] + .TMP_LEN;
1782      IF .LOCAL_OBJTYP EQL ACLSC_FILE
1783      THEN
1784      BEGIN
1785      RESNAM_TEXT[RSN_W_FID_NUM] = .FILE_FIB[FIBSW_FID_NUM];
1786      RESNAM_TEXT[RSN_W_FID_SEQ] = .FILE_FIB[FIBSW_FID_SEQ];
1787      LOCK_RESNAM[DSCSW_LENGTH] = .LOCK_RESNAM[DSCSW_LENGTH] + 4;
1788      END;
1789      END;
1790
1791      END;
```

```
1794 1789 2 ! For files, process the attribute list, and pass it through to the ACP.
1795 1790 3 ! for all other objects, the attribute processing is done here.
1796 1791 3
1797 1792 2 IF .LOCAL_OBJTYP EQL ACLSC_FILE
1798 1793 2 THEN
1799 1794 2 BEGIN
1800 1795 2
1801 1796 2 ! Build the ACP attribute list.
1802 1797 2
1803 1798 2 STATUS = LIB$GET_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
1804 1799 2 IF NOT .STATUS
1805 1800 2 THEN
1806 1801 2 BEGIN
1807 1802 2 IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1808 1803 2 RETURN .STATUS;
1809 1804 2 END;
1810 1805 2 FUNCTION CODE = IOS_ACCESS;
1811 1806 2 ACP_ATR_PTR = 0;
1812 1807 2 INCR J FROM 0 TO .ITEM_COUNT - 1
1813 1808 2 DO
1814 1809 2 BEGIN
1815 1810 2 IF PROBER (%REF (0), %REF (ITMSW_ITEM), ITEM_LIST[J, 0,0,0,0])
1816 1811 2 THEN
1817 1812 2 BEGIN
1818 1813 2 ITEM_CODE = .ITEM_LIST[J, ITMSW_ITEMCODE];
1819 1814 2 ITEM_SIZE = .ITEM_LIST[J, ITMSW_BUFSIZ];
1820 1815 2 ITEM_ADDR = .ITEM_LIST[J, ITMSW_BUFADR];
1821 1816 2 END
1822 1817 2 ELSE
1823 1818 2 BEGIN
1824 1819 2 STATUS = SSS_ACCVIO;
1825 1820 2 LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
1826 1821 2 IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1827 1822 2 RETURN .STATUS;
1828 1823 2 END;
1829 1824 2 IF .ITEM_CODE GTR MAX_ACL_ATR
1830 1825 2 THEN
1831 1826 2 BEGIN
1832 1827 2 STATUS = SSS_BADPARAM;
1833 1828 2 LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
1834 1829 2 IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1835 1830 2 RETURN .STATUS;
1836 1831 2 END;
1837 1832 2
1838 1833 2 IF .ITEM_CODE EQL ACLSC_RLOCK_ACL
1839 1834 2 OR .ITEM_CODE EQL ACLSC_WLOCK_ACL
1840 1835 2 THEN
1841 1836 2 BEGIN
1842 1837 2 IF .ITEM_SIZE LSSU 4
1843 1838 2 THEN
1844 1839 2 BEGIN
1845 1840 2 STATUS = SSS_BADPARAM;
1846 1841 2 LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
1847 1842 2 IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1848 1843 2 RETURN .STATUS;
1849 1844 2 END;
1850 1845 2 IF NOT EXESPROBER (0, .ITEM_SIZE, .ITEM_ADDR)
```



```
1851 1846 5 THEN
1852 1847 6 BEGIN
1853 1848 6 STATUS = $$$ ACCVIO;
1854 1849 6 LIB$FREE_VM (%REF ((.ITEM COUNT + 1) * 8), ACP_ATR_LIST);
1855 1850 6 IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1856 1851 6 RETURN .STATUS;
1857 1852 6 END;
1858 1853 5 STATUS = $ENQ (LKMODE = (IF .ITEM_CODE EQL ACL$C_RLOCK_ACL
1859 1854 5 THEN LCK$K_CRMODE ELSE [CK$K_PWMODE),
1860 1855 5 LKSB = LOCAL_IOSB,
1861 1856 5 RESNAM = LOCK_RESNAM,
1862 1857 5 PARID = (IF .CALL_ACMODE EQL PSL$C_USER
1863 1858 5 THEN .PARENT_ID
1864 1859 5 ELSE 0),
1865 1860 5 FLAGS = LCK$M_NOQUEUE OR
1866 1861 5 LCK$M_SYNCSTS OR
1867 1862 5 LCK$M_SYSTEM,
1868 1863 5 ACMODE = PSL$C_USER);
1869 1864 5 IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
1870 1865 5 IF NOT .STATUS
1871 1866 5 THEN
1872 1867 6 BEGIN
1873 1868 6 LIB$FREE_VM (%REF ((.ITEM COUNT + 1) * 8), ACP_ATR_LIST);
1874 1869 6 IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1875 1870 6 RETURN .STATUS;
1876 1871 6 END;
1877 1872 5 CH$COPY (4, LOCAL_IOSB[2],
1878 1873 5 0,
1879 1874 5 .ITEM_SIZE, .ITEM_ADDR); ! Copy lock-id
1880 1875 5 END
1881 1876 5 ELSE IF .ITEM_CODE EQL ACL$C_UNLOCK_ACL
1882 1877 4 THEN
1883 1878 4 BEGIN
1884 1879 4 IF .ITEM_SIZE LSSU 4
1885 1880 4 THEN
1886 1881 5 BEGIN
1887 1882 6 STATUS = $$$ BADPARAM;
1888 1883 6 LIB$FREE_VM (%REF ((.ITEM COUNT + 1) * 8), ACP_ATR_LIST);
1889 1884 6 IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1890 1885 6 RETURN .STATUS;
1891 1886 6 END;
1892 1887 5 IF NOT EX$PROBER (0, .ITEM_SIZE, .ITEM_ADDR)
1893 1888 5 THEN
1894 1889 6 BEGIN
1895 1890 6 STATUS = $$$ ACCVIO;
1896 1891 6 LIB$FREE_VM (%REF ((.ITEM COUNT + 1) * 8), ACP_ATR_LIST);
1897 1892 6 IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1898 1893 6 RETURN .STATUS;
1899 1894 6 END;
1900 1895 5 CH$COPY (.ITEM_SIZE, .ITEM_ADDR, 0, 4, LOCAL_LOCKID);
1901 1896 5 END
1902 1897 5 ELSE
1903 1898 4 BEGIN
1904 1899 4
1905 1900 4
1906 1901 4 ! Save the converted attribute code and other information.
1907 1902 4
```

```
1908 1903 5      ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_TYPE] = .ACL_TO_ATR_TAB[.ITEM_CODE];
1909 1904 5      IF .ITEM_CODE EQL ACL$C_ADDACLENT OR .ITEM_CODE EQL ACL$C_DELEACLENT
1910 1905 5      OR .ITEM_CODE EQL ACL$C_MODACLENT OR .ITEM_CODE EQL ACL$C_DELETEACL
1911 1906 5      THEN FUNCTION_CODE = IOS$MODIFY;
1912 1907 5      ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_SIZE] = .ITEM_SIZE;
1913 1908 5      ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_ADDR] = .ITEM_ADDR;
1914 1909 5      ACP_ATR_PTR = .ACP_ATR_PTR + 1;
1915 1910 5      END;
1916 1911 5      END;
1917 1912 5
1918 1913 5      ! Tie off the attribute descriptor list.
1919 1914 5
1920 1915 5      ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_TYPE] = 0;
1921 1916 5      ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_SIZE] = 0;
1922 1917 5
1923 1918 5      ! Initialize the FIB, and call the ACP to process the attribute list.
1924 1919 5
1925 1920 5      FILE_FIB[FIB$S_ACLCTX] = .ACL_CONTEXT;
1926 1921 5      STATUS = $QIOW (CHAN = .IO_CHANNEL,
1927 1922 5          FUNC = .FUNCTION_CODE,
1928 1923 5          IOSB = LOCAL_IOSB,
1929 1924 5          P1 = FILE_FIB_DESC,
1930 1925 5          P5 = .ACP_ATR_LIST);
1931 1926 5      IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
1932 1927 5      IF .STATUS THEN STATUS = .FILE_FIB[FIB$S_ACL_STATUS];
1933 1928 5      IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1934 1929 5
1935 1930 5      STATUS2 = LIB$FREE_VM (XREF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
1936 1931 5      IF .STATUS AND NOT .STATUS2 THEN STATUS = .STATUS2;
1937 1932 5
1938 1933 5      ! If an unlock request was made on the file's ACL, do it now.
1939 1934 5
1940 1935 5      IF .LOCAL_LOCKID NEQ 0 THEN STATUS = $DEQ (LKID = .LOCAL_LOCKID);
1941 1936 5      END
1942 1937 5
1943 1938 5      ! For non-file objects, the queue head has been located; loop through
1944 1939 5      ! the item list performing the actions specified.
1945 1940 5
1946 1941 5      ELSE
1947 1942 5      BEGIN
1948 1943 5          CMK_ARG_LIST[0] = 3;
1949 1944 5          CMK_ARG_LIST[1] = .ITEM_COUNT;
1950 1945 5          CMK_ARG_LIST[2] = .ITEM_LIST;
1951 1946 5          CMK_ARG_LIST[3] = .SHARE;
1952 1947 5          STATUS = $CMKRN (ROUTIN = ACL_DISPATCH,
1953 1948 5              ARG1ST = CMK_ARG_LIST);
1954 1949 5      END;
1955 1950 5
1956 1951 5      ! If necessary, deassign the channel assigned.
1957 1952 5
1958 1953 5      IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
1959 1954 5
1960 1955 5      ! If necessary, return the context.
1961 1956 5
1962 1957 5      IF .CONTEXT NEQ 0
1963 1958 5      THEN IF PROBEW (XREF (0), XREF (4), .CONTEXT)
1964 1959 5      THEN
```

```
1965 1960 3 BEGIN
1966 1961 3 IF .LOCAL OBJTYP EQL ACL$C FILE
1967 1962 3 THEN .CONTEXT = .FILE_FIB[FIB$$_ACLCTX]
1968 1963 3 ELSE .CONTEXT = .ACL_CONTEXT;
1969 1964 3 END
1970 1965 2 ELSE STATUS = $$$_ACCVIO;
1971 1966 2
1972 1967 2 RETURN .STATUS;
1973 1968 2
1974 1969 1 END;
INFO#250 L1:1779
Referenced LOCAL symbol TMP_LEN is probably not initialized
```

! End of routine SYSSCHANGE_ACL

```
00000023 00000022 00000021 00000020 0000001F 00000000 0054C P.AFL: .PSECT $SPLITS,NOWRT,NOEXE,2
00000027 00000026 00000025 00000024 00564 .LONG 0, 31, 32, 33, 34, 35, 36, 37, 38, 39

.EXTRN SYSSCMKRN, SYSSOPEN
.EXTRN SYSSQIOW, SYSSASSIGN
.EXTRN SYSSDASSGN, SYSSGETDVI
.EXTRN SYSSENO, SYSSDEQ

.PSECT $CODE$,NOWRT,2
.ENTRY SYSSCHANGE_ACL, Save R2,R3,R4,R5,R6,R7,R8,- 1391
R9,R10,R11
MOVAB -920(SP), SP
MOVCS #40, P.AFL, #0, #52, ACL_TO_ATR_TAB 1486

CLRL CHANGE_ACMODE 1493
MOVL ACCESS_MODE, R0 1494
TSTL R0
BEQL 1$
PROBER #0, #4, (R0) 1495
BEQL 5$
MOVL (R0), CHANGE_ACMODE 1496
MOVPSL PSL 1499
EXTZV #22, #2, PSL, CALL_ACMODE 1500
MOVL CHANGE_ACMODE, R0 1501
CMLP R0, CALL_ACMODE
BGEQU 2$
MOVL CALL_ACMODE, R0
MOVL R0, CHANGE_ACMODE
CMLP CHANGE_ACMODE, #3 1505
BGRU 6$
MOVZWL CHANNEL, R0 1509
MOVW R0, LOCAL_CHANNEL
MOVW R0, IO_CHANNEL
MOVZWL IO_CHANNEL, R8 1510
BEQL 3$
MOVL R8, R0 1513
JSB IOCSVERIFYCHAN
MOVL R0, STATUS
BLBS STATUS, 3$ 1514
```

| | | | | | | | | | |
|------|-----------|-----------|------|-------|-------|-------|--------|-----------------------|------|
| | | | 059B | 31 | 00080 | BRW | 81\$ | | |
| | 50 | 08 | AC | D0 | 00083 | 3\$: | MOVL | OBJECT_TYPE, R0 | 1519 |
| | | | 03 | 12 | 00087 | | BNEQ | 4\$ | |
| | | | 0293 | 31 | 00089 | | BRW | 36\$ | |
| 60 | 04 | | 00 | 0C | 0008C | 4\$: | PROBER | #0, #4, (R0) | 1520 |
| | | | 6D | 13 | 00090 | 5\$: | BEQL | 12\$ | |
| | 56 | | 60 | D0 | 00092 | | MOVL | (R0), LOCAL_OBJTYP | 1521 |
| | | | 05 | 13 | 00095 | | BEQL | 6\$ | 1527 |
| | 07 | | 56 | D1 | 00097 | | CMPL | LOCAL_OBJTYP, #7 | 1528 |
| | | | 04 | 1B | 0009A | | BLEQU | 7\$ | |
| | 50 | | 14 | D0 | 0009C | 6\$: | MOVL | #20, R0 | 1529 |
| | | | | 04 | 0009F | | RET | | |
| | 50 | 0C | AC | D0 | 000A0 | 7\$: | MOVL | OBJECT_NAME, R0 | 1533 |
| | | | 24 | 13 | 000A4 | | BEQL | 8\$ | |
| 60 | 08 | | 00 | 0C | 000A6 | | PROBER | #0, #8, (R0) | 1536 |
| | | | 53 | 13 | 000AA | | BEQL | 12\$ | |
| F0 | AD | | 60 | 3C | 000AC | | MOVZWL | (R0), OBJECT_DESC | 1538 |
| F4 | AD | 04 | A0 | D0 | 000B0 | | MOVL | 4(R0), OBJECT_DESC+4 | 1539 |
| | 50 | F4 | AD | D0 | 000B5 | | MOVL | OBJECT_DESC+4, R0 | 1540 |
| | 51 | F0 | AD | D0 | 000B9 | | MOVL | OBJECT_DESC, R1 | |
| | | | 53 | D4 | 000BD | | CLRL | R3 | |
| | 05 | 00000000G | 00 | 16 | 000BF | | JSB | EXESPROBER | |
| | | | 50 | E8 | 000C5 | | BLBS | R0, 9\$ | |
| | | | 62 | 11 | 000C8 | | BRB | 15\$ | 1541 |
| | | F0 | AD | 7C | 000CA | 8\$: | CLRB | OBJECT_DESC | 1545 |
| | | 00000000' | EF | D4 | 000CD | 9\$: | CLRL | ACL_CONTEXT | 1551 |
| | | 1C | AE | D4 | 000D3 | | CLRL | 28(SP) | 1552 |
| | | 1C | AC | D5 | 000D6 | | TSTL | CONTEXT | |
| | | | 12 | 13 | 000D9 | | BEQL | 10\$ | |
| | | 1C | AE | D6 | 000DB | | INCL | 28(SP) | |
| 1C | BC | 04 | 00 | 0D | 000DE | | PROBER | #0, #4, @CONTEXT | 1553 |
| | | | 47 | 13 | 000E3 | | BEQL | 15\$ | |
| | 00000000' | EF | 1C | BC | D0 | 000E5 | MOV | @CONTEXT, ACL_CONTEXT | 1554 |
| | | 5A | 01 | 90 | 000ED | 10\$: | MOVB | #1, SHARE | 1559 |
| | | 59 | 10 | AC | D0 | 000F0 | MOVL | ITEM_LIST, R9 | 1561 |
| | | | 50 | D4 | 000F4 | | CLRL | J | |
| 51 | | 50 | 0C | C5 | 000F6 | 11\$: | MULL3 | #12, J, R1 | |
| 6149 | | 0C | 00 | 0C | 000FA | | PROBER | #0, #12, (R1)[R9] | |
| | | | 2B | 13 | 000FF | 12\$: | BEQL | 15\$ | |
| | | 6149 | 9F | 00101 | | | PUSHAB | (R1)[R9] | 1562 |
| | | | 9E | B5 | 00104 | | TSTW | @(SP)+ | |
| | | | 05 | 12 | 00106 | | BNEQ | 13\$ | |
| | | 57 | 50 | D0 | 00108 | | MOVL | J, ITEM_COUNT | 1565 |
| | | | 2B | 11 | 0010B | | BRB | 17\$ | 1564 |
| | | 02 | A149 | 9F | 0010D | 13\$: | PUSHAB | 2(R1)[R9] | 1570 |
| | | | 9E | 3C | 00111 | | MOVZWL | @(SP)+, R1 | |
| | | 51 | 51 | B1 | 00114 | | CMPL | R1, #1 | |
| | | | 0F | 13 | 00117 | | BEQL | 14\$ | |
| | | 02 | 51 | B1 | 00119 | | CMPL | R1, #2 | 1571 |
| | | | 0A | 13 | 0011C | | BEQL | 14\$ | |
| | | 03 | 51 | B1 | 0011E | | CMPL | R1, #3 | 1572 |
| | | | 05 | 13 | 00121 | | BEQL | 14\$ | |
| | | 06 | 51 | B1 | 00123 | | CMPL | R1, #6 | 1573 |
| | | | 08 | 12 | 00126 | | BNEQ | 16\$ | |
| | | | 5A | 94 | 00128 | 14\$: | CLRB | SHARE | 1574 |
| | | | 04 | 11 | 0012A | | BRB | 16\$ | 1562 |
| | | 50 | 0C | D0 | 0012C | 15\$: | MOVL | #12, R0 | 1576 |

| | | | | | | | | | | |
|------|-----------|------|-----------|------|----------|-------|-------|--------|-----------------------------------|------|
| 18 | BE | 50 | 7FFFFFFF | 8F | F3 | 0012F | 16\$: | RET | | 1561 |
| | 00 | 6E | | 00 | 2C | 00130 | 17\$: | AOBLEQ | #2147483647, J, 11\$ | 1580 |
| 08 | | | 78 | AE | 0C | 00138 | | MOVCS | #0, (SP), #0, #24, DVI_ATR_LIST | |
| | 00 | 6E | 00000000 | 00 | 2C | 0013F | | MOVCS | #0, (SP), #0, #8, LOCK_RESNAM | 1581 |
| | | | 28 | EF | 00 | 00144 | | CLRL | LOCAL_LOCKID | 1582 |
| | 00000000 | EF | 00000000 | 08 | D4 | 00149 | | MOVW | #8, LOCK_RESNAM | 1586 |
| | 00000000 | EF | 00000000 | 08 | B0 | 0014C | | MOVAB | RESNAM_TEXT, LOCK_RESNAM+4 | 1587 |
| 08 | 00 | 04 | 00000000 | EF | 9E | 00153 | | MOVL | LOCK_PREFIX[LOCAL_OBJTYP], R0 | 1588 |
| | | | 00000000 | 50 | D0 | 0015E | | MOVCS | (R0), #4(R0), #0, #8, RESNAM_TEXT | |
| | | | 00000000 | 60 | 2C | 00166 | | | | |
| | | 03 | 00000000 | EF | 00 | 0016C | | CMPL | CALL_ACMODE, #3 | 1596 |
| | | | 00000000 | 20 | 12 | 00171 | | BNEQ | 18\$ | |
| | | | 00000000 | EF | D5 | 00178 | | TSTL | PARENT_ID | 1598 |
| | | | | 18 | 12 | 0017A | | BNEQ | 18\$ | |
| | | | 00000000V | 7E | D4 | 00180 | | CLRL | -(SP) | 1601 |
| | 00000000G | 00 | | EF | 9F | 00182 | | PUSHAB | GET_PARENT_LOCK | |
| | | 5B | | 02 | FB | 00184 | | CALLS | #2, SYSSCMRNL | |
| | | 03 | | 50 | D0 | 0018A | | MOVL | R0, STATUS | |
| | | | | 5B | E8 | 00191 | | BLBS | STATUS, 18\$ | 1602 |
| | 06 | 01 | | 0484 | 31 | 00194 | | BRW | 81\$ | |
| 0131 | 012C | 00F7 | | 56 | CF | 00197 | 18\$: | CASEL | LOCAL_OBJTYP, #1, #6 | 1610 |
| | 017C | 0163 | | 0011 | | 0019E | 19\$: | .WORD | 20\$-19\$,- | |
| | | | | 014A | | 001A6 | | | 26\$-19\$,- | |
| | | | | | | | | | 29\$-19\$,- | |
| | | | | | | | | | 31\$-19\$,- | |
| | | | | | | | | | 32\$-19\$,- | |
| | | | | | | | | | 33\$-19\$,- | |
| | | | | | | | | | 35\$-19\$ | |
| | | | | | | | | | 29\$ | |
| 0040 | 8F | 00 | | 011B | 31 | 001AC | 20\$: | BRW | | 1741 |
| | | | | 00 | 2C | 001AF | | MOVCS | #0, (SP), #0, #64, FILE_FIB | 1617 |
| 08 | | 00 | 0090 | CE | | 001B6 | | | | |
| | | | 00D0 | 00 | 2C | 001B9 | | MOVCS | #0, (SP), #0, #8, FILE_FIB_DESC | 1618 |
| | | | 00D4 | CE | | 001BE | | | | |
| | | | 00BE | CE | 00000000 | 001C1 | | MOVZBW | #64, FILE_FIB_DESC | 1619 |
| | | | | 8F | 9B | 001C7 | | MOVAB | FILE_FIB, FILE_FIB_DESC+4 | 1620 |
| | | | | CE | 9E | 001CE | | MOVW | CHANGE_ACMODE, FILE_FIB+46 | 1621 |
| | | | | EF | 90 | 001D7 | | MOVB | R8 | 1625 |
| | | | | 58 | D5 | 001D9 | | TSTL | 21\$ | |
| | | | | 03 | 13 | 001DB | | BEQL | 24\$ | |
| 0050 | 8F | 00 | | 0087 | 31 | 001DE | 21\$: | BRW | | |
| | | | | 00 | 2C | 001E5 | | MOVCS | #0, (SP), #0, #80, \$RMS_PTR | 1632 |
| | | | A0 | AD | | 001E7 | | | | |
| | | | 5003 | 8F | B0 | 001ED | | MOVW | #20483, \$RMS_PTR | |
| | A4 | AD | 00020000 | 8F | D0 | 001F5 | | MOVL | #131072, \$RMS_PTR+4 | |
| | B6 | AD | | 02 | 90 | 001F9 | | MOVB | #2, \$RMS_PTR+22 | |
| | BF | AD | | 02 | 90 | 001FD | | MOVB | #2, \$RMS_PTR+31 | |
| | C8 | AD | FF40 | CD | 9E | 00203 | | MOVAB | FILE_NAME, \$RMS_PTR+40 | |
| | CC | AD | F4 | AD | D0 | 00208 | | MOVL | OBJECT_DESC+4, \$RMS_PTR+44 | |
| | D4 | AD | F0 | AD | 90 | 0020D | | MOVB | OBJECT_DESC, \$RMS_PTR+52 | |
| 0060 | 8F | 00 | | 00 | 2C | 00214 | | MOVCS | #0, (SP), #0, #96, \$RMS_PTR | 1637 |
| | | | FF40 | CD | | 00217 | | | | |
| | | | 6002 | 8F | B0 | 0021E | | MOVW | #24578, \$RMS_PTR | |
| | | | | 01 | 8E | 00223 | | MNEGB | #1, \$RMS_PTR+2 | |
| | | | 00D8 | CE | 9E | 0022A | | MOVAB | FILE_RES_NAME, \$RMS_PTR+4 | |
| | | | | 01 | 8E | 0022F | | MNEGB | #1, \$RMS_PTR+10 | |
| | | | FE40 | CD | 9E | | | MOVAB | FILE_EXP_NAME, \$RMS_PTR+12 | |

| | | | | | | | | | | | | | |
|----|----|----|-----------|----------|-----------|----|------|-------|-------|--------|------------------------------------|-----------------------------|------|
| EA | AD | 02 | B6 | AD | 4302 | 08 | 5A | E9 | 00236 | BLBC | SHARE, 22\$ | 1638 | |
| | | | B6 | AD | 2003 | AD | 8F | B0 | 00239 | MOVW | #17154, FILE_FAB+22 | 1642 | |
| | | | 04 | 00000000 | A0 | 06 | 11 | 0023F | BRB | 23\$ | | 1638 | |
| | | | 00000000G | 00 | | 8F | B0 | 00241 | 22\$: | MOVW | #8195, FILE_FAB+22 | 1647 | |
| | | | 20 | AE | AC | EF | F0 | 00247 | 23\$: | INSV | CHANGE_ACMODE, #4, #2, FILE_FAB+74 | 1649 | |
| | | | 5B | | | AD | 9F | 00251 | | PUSHAB | FILE_FAB | 1651 | |
| | | | 45 | | | 01 | FB | 00254 | | CALLS | #1, SYSSOPEN | | |
| | | | | | | 50 | DO | 0025B | | MOVL | R0, STATUS | | |
| | | | | | | AD | B0 | 0025E | | MOVW | FILE_FAB+12, IO_CHANNEL | 1652 | |
| | | | | | | 03 | 11 | 00263 | | BRB | 25\$ | 1625 | |
| | | | | | | 01 | DO | 00265 | 24\$: | MOVL | #1, STATUS | 1654 | |
| | | | | | | 5B | E9 | 00268 | 25\$: | BLBC | STATUS, 27\$ | 1659 | |
| | | | | | | 7E | 7C | 0026B | | CLRQ | -(SP) | 1665 | |
| | | | | | | 7E | 7C | 0026D | | CLRQ | -(SP) | | |
| | | | | | | 7E | D4 | 0026F | | CLRL | -(SP) | | |
| | | | | | 00E4 | CE | 9F | 00271 | | PUSHAB | FILE_FIB_DESC | | |
| | | | | | F8 | 7E | 7C | 00275 | | CLRQ | -(SP) | | |
| | | | | | | AD | 9F | 00277 | | PUSHAB | LOCAL_IOSB | | |
| | | | | | | 32 | DD | 0027A | | PUSHL | #50 | | |
| | | | | | 7E | 48 | AE | 3C | 0027C | MOVZWL | IO_CHANNEL, -(SP) | | |
| | | | | | | 7E | D4 | 00280 | | CLRL | -(SP) | | |
| | | | | | 00000000G | 00 | OC | FB | 00282 | CALLS | #12, SYSSQIOW | | |
| | | | | | | 5B | DO | 00289 | | MOVL | R0, STATUS | | |
| | | | | | | 21 | 5B | E9 | 0028C | BLBC | STATUS, 27\$ | 1666 | |
| | | | | | | 5B | F8 | AD | 3C | MOVZWL | LOCAL_IOSB, STATUS | | |
| | | | | | | | 38 | 11 | 00293 | BRB | 30\$ | 1610 | |
| | | | | | | | 58 | D5 | 00295 | TSTL | R8 | 1675 | |
| | | | | | | | 17 | 12 | 00297 | BNEQ | 27\$ | | |
| | | | | | | | AD | D5 | 00299 | TSTL | OBJECT_DESC | 1678 | |
| | | | | | | | 66 | 13 | 0029C | BEQL | 34\$ | | |
| | | | | | | | 7E | 7C | 0029E | CLRQ | -(SP) | 1681 | |
| | | | | | | 28 | AE | 9F | 002A0 | PUSHAB | IO_CHANNEL | | |
| | | | | | | F0 | AD | 9F | 002A3 | PUSHAB | OBJECT_DESC | | |
| | | | | | | | 04 | FB | 002A6 | CALLS | #4, SYSSASSIGN | | |
| | | | | | | | 50 | DO | 002AD | MOVL | R0, STATUS | | |
| | | | | | | | 5B | E8 | 002B0 | BLBS | STATUS, 28\$ | 1686 | |
| | | | | | | | 01F9 | 31 | 002B3 | BRW | 60\$ | | |
| | | | | | | | 01 | DO | 002B6 | 27\$: | MOVW | #1, CMK_ARG_LIST | 1689 |
| | | | | | | | AE | 3C | 002BA | 28\$: | MOVZWL | IO_CHANNEL, CMK_ARG_LIST+4 | 1690 |
| | | | | | | | AE | 9F | 002BF | | PUSHAB | CMK_ARG_LIST | 1692 |
| | | | | | | | EF | 9F | 002C2 | | PUSHAB | GET_UCB_ACL | |
| | | | | | | | 6D | 11 | 002C8 | BRB | 38\$ | | |
| | | | | | | | 14 | DO | 002CA | 29\$: | MOVL | #20, STATUS | 1698 |
| | | | | | | | 72 | 11 | 002CD | 30\$: | BRB | 39\$ | 1610 |
| | | | | | | | AD | D5 | 002CF | 31\$: | TSTL | OBJECT_DESC | 1703 |
| | | | | | | | 4B | 13 | 002D2 | | BEQL | 36\$ | |
| | | | | | | | 01 | DO | 002D4 | | MOVL | #1, CMK_ARG_LIST | 1705 |
| | | | | | | | AD | 9E | 002D8 | | MOVAB | OBJECT_DESC, CMK_ARG_LIST+4 | 1706 |
| | | | | | | | AE | 9F | 002DD | | PUSHAB | CMK_ARG_LIST | 1708 |
| | | | | | | | EF | 9F | 002E0 | | PUSHAB | GET_CEB_ACL | |
| | | | | | | | 4F | 11 | 002E6 | BRB | 38\$ | | |
| | | | | | | | AD | D5 | 002E8 | 32\$: | TSTL | OBJECT_DESC | 1713 |
| | | | | | | | 32 | 13 | 002EB | | BEQL | 36\$ | |
| | | | | | | | 01 | DO | 002ED | | MOVL | #1, CMK_ARG_LIST | 1715 |
| | | | | | | | AD | 9E | 002F1 | | MOVAB | OBJECT_DESC, CMK_ARG_LIST+4 | 1716 |
| | | | | | | | AE | 9F | 002F6 | | PUSHAB | CMK_ARG_LIST | 1718 |
| | | | | | | | EF | 9F | 002F9 | | PUSHAB | GET_LNT_ACL | |

| | | | | | | | | | |
|-----------|----|-----------|------|-------|-------|--------|-------------------------------|------|--|
| | | | 36 | 11 | 002FF | BRB | 38\$ | | |
| | | FO | AD | D7 | 00301 | TSTL | OBJECT_DESC | 1723 | |
| | | | 19 | 13 | 00304 | BEQL | 36\$ | | |
| 30 | AE | | 01 | D0 | 00306 | MOVL | #1, CMK_ARG_LIST | 1725 | |
| 34 | AE | | AD | 9E | 0030A | MOVAB | OBJECT_DESC, CMK_ARG_LIST+4 | 1726 | |
| | | 30 | AE | 9F | 0030F | PUSHAB | CMK_ARG_LIST | 1728 | |
| | | 00000000V | EF | 9F | 00312 | PUSHAB | GET_PCB_ACL | | |
| | | | 1D | 11 | 00318 | BRB | 38\$ | | |
| | | FO | AD | D5 | 0031A | TSTL | OBJECT_DESC | 1733 | |
| | | | 06 | 12 | 0031D | BNEQ | 37\$ | | |
| | 50 | 0114 | 8F | 3C | 0031F | MOVZWL | #276, R0 | 1734 | |
| | | | | 04 | 00324 | RET | | | |
| 30 | AE | | 01 | D0 | 00325 | MOVL | #1, CMK_ARG_LIST | 1735 | |
| 34 | AE | | AD | 9E | 00329 | MOVAB | OBJECT_DESC, CMK_ARG_LIST+4 | 1736 | |
| | | 30 | AE | 9F | 0032E | PUSHAB | CMK_ARG_LIST | 1738 | |
| | | 00000000V | EF | 9F | 00331 | PUSHAB | GET_GBL_ACL | | |
| 00000000G | 00 | | 02 | FB | 00337 | CALLS | #2, SYS\$CMKRNL | | |
| | 5B | | 50 | D0 | 0033E | MOVL | R0, STATUS | | |
| | 4C | | 5B | E9 | 00341 | BLBC | STATUS, 42\$ | 1746 | |
| | | 18 | AE | D4 | 00344 | CLRL | 24(SP) | 1757 | |
| | 01 | | 56 | D1 | 00347 | CMPL | LOCAL_OBJTYP, #1 | | |
| | | | 05 | 12 | 0034A | BNEQ | 40\$ | | |
| | | 18 | AE | D6 | 0034C | INCL | 24(SP) | | |
| | | | 05 | 11 | 0034F | BRB | 41\$ | | |
| | 02 | | 56 | D1 | 00351 | CMPL | LOCAL_OBJTYP, #2 | | |
| | | | 59 | 12 | 00354 | BNEQ | 43\$ | | |
| 78 | AE | 00F00017 | 8F | D0 | 00356 | MOVL | #15728663, DVI_ATR_LIST | 1765 | |
| 7C | AE | 00000000 | EF | 9E | 0035E | MOVAB | RESNAM_TEXT+8, DVI_ATR_LIST+4 | 1766 | |
| 0080 | CE | 24 | AE | 9E | 00366 | MOVAB | TMP_LEN, DVI_ATR_LIST+8 | 1767 | |
| | | | 7E | 7C | 0036C | CLRQ | -(SP) | 1770 | |
| | | | 7E | D4 | 0036E | CLRL | -(SP) | | |
| | | F8 | AD | 9F | 00370 | PUSHAB | LOCAL_IOSB | | |
| | | 0088 | CE | 9F | 00373 | PUSHAB | DVI_ATR_LIST | | |
| | | | 7E | D4 | 00377 | CLRL | -(SP) | | |
| | 7E | 38 | AE | 3C | 00379 | MOVZWL | IO_CHANNEL, -(SP) | | |
| | | | 7E | D4 | 0037D | CLRL | -(SP) | | |
| 00000000G | 00 | | 08 | FB | 0037F | CALLS | #8, SYS\$GETDVI | | |
| | 5B | | 50 | D0 | 00386 | MOVL | R0, STATUS | | |
| | 48 | | 5B | E9 | 00389 | BLBC | STATUS, 45\$ | 1771 | |
| | 5B | F8 | AD | 3C | 0038C | MOVZWL | LOCAL_IOSB, STATUS | | |
| | 41 | | 5B | E9 | 00390 | BLBC | STATUS, 45\$ | 1772 | |
| 00000000 | EF | 24 | AE | A0 | 00393 | ADDW2 | TMP_LEN, LOCK_RESNAM | 1779 | |
| | 10 | 18 | AE | E9 | 0039B | BLBC | 24(SP), 43\$ | 1780 | |
| 00000000 | EF | 0094 | CE | D0 | 0039F | MOVL | FILE_FIB+4, RESNAM_TEXT+24 | 1783 | |
| 00000000 | EF | | 04 | A0 | 003A8 | ADDW2 | #4, LOCK_RESNAM | 1785 | |
| | 03 | 18 | AE | E8 | 003AF | BLBS | 24(SP), 44\$ | 1798 | |
| | | | 0211 | 31 | 003B3 | BRW | 75\$ | | |
| | | 2C | AE | 9F | 003B6 | PUSHAB | ACP_ATR_LIST | | |
| 08 | AE | | 03 | 78 | 003B9 | ASHL | #3, ITEM_COUNT, 8(SP) | | |
| | | | 08 | C0 | 003BE | ADDL2 | #8, 8(SPT) | | |
| | 08 | | AE | D0 | 003C2 | MOVL | 8(SP), 24(SP) | | |
| | 18 | | AE | 9F | 003C7 | PUSHAB | 24(SP) | | |
| 00000000G | 00 | | 02 | FB | 003CA | CALLS | #2, LIB\$GET_VM | | |
| | 5B | | 50 | D0 | 003D1 | MOVL | R0, STATUS | | |
| | 03 | | 5B | E8 | 003D4 | BLBS | STATUS, 46\$ | 1799 | |
| | | 00D5 | 31 | 003D7 | BRW | 60\$ | | | |
| | | | 32 | D0 | 003DA | MOVL | #50, FUNCTION_CODE | 1805 | |

| | | | | | | | | | |
|------|-----------|-----------|------|------|-------|--------|-----------|---|------|
| | | | | 58 | D4 | 003DE | CLRL | ACP_ATR_PTR | 1806 |
| | | | | 01 | CE | 003E0 | MNEGL | #1, J | 1807 |
| | | | | 014D | 31 | 003E3 | BRW | 69\$ | |
| | 50 | | | 0C | C5 | 003E6 | MULL3 | #12, J, R0 | 1810 |
| 6049 | | | | 00 | 0C | 003EA | PROBER | #0, #12, (R0)[R9] | |
| | | | | 18 | 13 | 003EF | BEQL | 48\$ | |
| | | 02 | A049 | 9F | 003F1 | PUSHAB | 2(R0)[R9] | | 1813 |
| | | | | 9E | 3C | 003F5 | MOVZWL | @(SP)+, ITEM_CODE | |
| | | | | 6049 | 9F | 003F8 | PUSHAB | (R0)[R9] | 1814 |
| 08 | AE | | | 9E | 3C | 003FB | MOVZWL | @(SP)+, ITEM_SIZE | |
| | | 04 | A049 | 9F | 003FF | PUSHAB | 4(R0)[R9] | | 1815 |
| 0C | AE | | | 9E | D0 | 00403 | MOVL | @(SP)+, ITEM_ADDR | |
| | | | | 06 | 11 | 00407 | BRB | 49\$ | 1810 |
| | | | | 0C | D0 | 00409 | MOVL | #12, STATUS | 1819 |
| | | | | 008E | 31 | 0040C | BRW | 59\$ | 1820 |
| | | | | 5A | D1 | 0040F | CMPL | ITEM_CODE, #12 | 1824 |
| | | | | 03 | 15 | 00412 | BLEQ | 50\$ | |
| | | | | 0083 | 31 | 00414 | BRW | 58\$ | |
| | | | | 54 | D4 | 00417 | CLRL | R4 | 1833 |
| 0A | | | | 5A | D1 | 00419 | CMPL | ITEM_CODE, #10 | |
| | | | | 04 | 12 | 0041C | BNEQ | 51\$ | |
| | | | | 54 | D6 | 0041E | INCL | R4 | |
| | | | | 05 | 11 | 00420 | BRB | 52\$ | |
| 08 | | | | 5A | D1 | 00422 | CMPL | ITEM_CODE, #11 | 1834 |
| | | | | 68 | 12 | 00425 | BNEQ | 57\$ | |
| 04 | 08 | | | AE | D1 | 00427 | CMPL | ITEM_SIZE, #4 | 1837 |
| | | | | 6D | 1F | 0042B | BLSSU | 58\$ | |
| 50 | | | | AE | D0 | 0042D | MOVL | ITEM_ADDR, R0 | 1845 |
| 51 | | | | AE | D0 | 00431 | MOVL | ITEM_SIZE, R1 | |
| | | | | 53 | D4 | 00435 | CLRL | R3 | |
| | | 00000000G | | 00 | 16 | 00437 | JSB | EXESPROBEW | |
| C9 | | | | 50 | E9 | 0043D | BLBC | R0, 48\$ | |
| 7E | | | | 03 | 7D | 00440 | MOVQ | #3, -(SP) | 1863 |
| | | | | 7E | 7C | 00443 | CLRQ | -(SP) | |
| | | | | 7E | D4 | 00445 | CLRL | -(SP) | |
| 03 | 00000000' | | | EF | D1 | 00447 | CMPL | CALL_ACMODE, #3 | |
| | | | | 08 | 12 | 0044E | BNEQ | 53\$ | |
| | 00000000' | | | EF | DD | 00450 | PUSHL | PARENT_ID | |
| | | | | 02 | 11 | 00456 | BRB | 54\$ | |
| | | | | 7E | D4 | 00458 | CLRL | -(SP) | |
| | 00000000' | | | EF | 9F | 0045A | PUSHAB | LOCK_RESNAM | |
| | | | | 1C | DD | 00460 | PUSHL | #28 | |
| | | | | AD | 9F | 00462 | PUSHAB | LOCAL_IOSB | |
| 04 | | | | 54 | E9 | 00465 | BLBC | R4, 55\$ | |
| | | | | 01 | DD | 00468 | PUSHL | #1 | |
| | | | | 02 | 11 | 0046A | BRB | 56\$ | |
| | | | | 04 | DD | 0046C | PUSHL | #4 | |
| | | | | 7E | D4 | 0046E | CLRL | -(SP) | |
| | 00000000G | | | 08 | FB | 00470 | CALLS | #11, SYSSENG | |
| | | | | 50 | D0 | 00477 | MOVL | R0, STATUS | |
| | | | | 5B | E9 | 0047A | BLBC | STATUS, 59\$ | 1864 |
| | | | | AD | 3C | 0047D | MOVZWL | LOCAL_IOSB, STATUS | |
| | | | | 5B | E9 | 00481 | BLBC | STATUS, 59\$ | 1865 |
| 08 | AE | | | 04 | 2C | 00484 | MOVCS | #4, LOCAL_IOSB+4, #0, ITEM_SIZE, @ITEM_ADDR | 1874 |
| | | | | 8E | | 0048B | | | |
| | | | | 6B | 11 | 0048D | BRB | 65\$ | 1833 |
| | | | | 0C | 5A | 0048F | CMPL | ITEM_CODE, #12 | 1877 |

| | | | | | | | | |
|-----------|-----------|-----------|------|-------|--------------|---|--|------|
| 04 | 08 | 68 | 12 | 00492 | BNEQ | 66\$ | | |
| | | AE | D1 | 00494 | CMPL | ITEM_SIZE, #4 | | 1880 |
| | | 1C | 1E | 00498 | BGEQU | 61\$ | | |
| 5B | | 14 | D0 | 0049A | 58\$: MOVL | #20, STATUS | | 1883 |
| | 2C | AE | 9F | 0049D | 59\$: PUSHAB | ACP_ATR_LIST | | 1884 |
| 04 | 08 | AE | D0 | 004A0 | MOVL | 8(SP), 4(SP) | | |
| | 04 | AE | 9F | 004A5 | PUSHAB | 4(SP) | | |
| 00000000G | 00 | 02 | FB | 004A8 | CALLS | #2, LIB\$FREE_VM | | |
| | 10 | AE | B5 | 004AF | 60\$: TSTW | LOCAL_CHANNEL | | 1885 |
| | | 2F | 13 | 004B2 | BEQL | 62\$ | | |
| | | 38 | 11 | 004B4 | BRB | 63\$ | | 1886 |
| 50 | 0C | AE | D0 | 004B6 | 61\$: MOVL | ITEM_ADDR, R0 | | 1888 |
| 51 | 08 | AE | D0 | 004BA | MOVL | ITEM_SIZE, R1 | | |
| | | 53 | D4 | 004BE | CLRL | R3 | | |
| | 00000000G | 00 | 16 | 004C0 | JSB | EXESPROBER | | |
| 28 | | 50 | E8 | 004C6 | BLBS | R0, 64\$ | | |
| 5B | | 0C | D0 | 004C9 | MOVL | #12, STATUS | | 1891 |
| | 2C | AE | 9F | 004CC | PUSHAB | ACP_ATR_LIST | | 1892 |
| 04 | 08 | AE | D0 | 004CF | MOVL | 8(SP), 4(SP) | | |
| | 04 | AE | 9F | 004D4 | PUSHAB | 4(SP) | | |
| 00000000G | 00 | 02 | FB | 004D7 | CALLS | #2, LIB\$FREE_VM | | |
| | 10 | AE | B5 | 004DE | TSTW | LOCAL_CHANNEL | | 1893 |
| | | 0B | 12 | 004E1 | BNEQ | 63\$ | | |
| | 20 | AE | 3C | 004E3 | 62\$: MOVZWL | IO_CHANNEL, -(SP) | | |
| 00000000G | 00 | 01 | FB | 004E7 | CALLS | #1, SYSSDASSGN | | |
| | | 012D | 31 | 004EE | 63\$: BRW | 81\$ | | 1894 |
| 04 | 00 | 0C | BE | 08 | 64\$: MOVCS | ITEM_SIZE, @ITEM_ADDR, #0, #4, LOCAL_LOCKID | | 1896 |
| | | 28 | AE | | | | | |
| | | 37 | 11 | 004FA | 65\$: BRB | 69\$ | | 1877 |
| | 50 | 2C | BE48 | 7E | 66\$: MOVAQ | @ACP_ATR_LIST[ACP_ATR_PTR], R0 | | 1903 |
| 02 | A0 | 44 | AE4A | F7 | CVTLW | ACL_TO_ATR_TAB[ITEM_CODE], 2(R0) | | |
| 01 | | | 5A | D1 | CMPL | ITEM_CODE, #1 | | 1904 |
| | | | 0F | 13 | BEQL | 67\$ | | |
| 02 | | | 5A | D1 | CMPL | ITEM_CODE, #2 | | |
| | | | 0A | 13 | BEQL | 67\$ | | |
| 03 | | | 5A | D1 | CMPL | ITEM_CODE, #3 | | 1905 |
| | | | 05 | 13 | BEQL | 67\$ | | |
| 06 | | | 5A | D1 | CMPL | ITEM_CODE, #6 | | |
| | | | 04 | 12 | BNEQ | 68\$ | | |
| 14 | AE | | 36 | D0 | 67\$: MOVL | #54, FUNCTION CODE | | 1906 |
| | | 2C | BE48 | 7F | 68\$: PUSHAQ | @ACP_ATR_LIST[ACP_ATR_PTR] | | 1907 |
| | 9E | 0C | AE | B0 | MOVW | ITEM_SIZE, @ (SP)+ | | |
| | 50 | 2C | BE48 | 7E | MOVAQ | @ACP_ATR_LIST[ACP_ATR_PTR], R0 | | 1908 |
| 04 | A0 | 0C | AE | D0 | MOVL | ITEM_ADDR, 4(R0) | | |
| | | | 58 | D6 | INCL | ACP_ATR_PTR | | 1909 |
| 02 | 56 | | 57 | F2 | 69\$: AOBLSS | ITEM_COUNT, J, 70\$ | | 1807 |
| | | | 03 | 11 | BRB | 71\$ | | |
| | | | FEAA | 31 | 70\$: BRW | 47\$ | | |
| | 50 | 2C | BE48 | 7E | 71\$: MOVAQ | @ACP_ATR_LIST[ACP_ATR_PTR], R0 | | 1915 |
| | | 02 | A0 | B4 | CLRW | 2(R0) | | |
| | | 2C | BE48 | 7F | PUSHAQ | @ACP_ATR_LIST[ACP_ATR_PTR] | | 1916 |
| | | | 9E | B4 | CLRW | @(SPT)+ | | |
| 00C0 | CE | 00000000' | EF | D0 | MOVL | ACL_CONTEXT, FILE_FIB+48 | | 1920 |
| | | | 7E | D4 | CLRL | -(SP) | | 1925 |
| | | 30 | AE | DD | PUSHL | ACP_ATR_LIST | | |
| | | | 7E | 7C | CLRW | -(SP) | | |
| | | | 7E | D4 | CLRL | -(SP) | | |

| | | | | | | | | |
|-----------|----|-----------|----|-------|-------|--------|----------------------------|------|
| | | 00E4 | CE | 9F | 0055C | PUSHAB | FILE_FIB_DESC | |
| | | | 7E | 7C | 00560 | CLRQ | -(SP) | |
| | | F8 | AD | 9F | 00562 | PUSHAB | LOCAL_IOSB | |
| | | 38 | AE | DD | 00565 | PUSHL | FUNCTION CODE | |
| | 7E | 48 | AE | 3C | 00568 | MOVZWL | IO_CHANNEL, -(SP) | |
| | | | 7E | D4 | 0056C | CLRL | -(SP) | |
| 00000000G | 00 | | OC | FB | 0056E | CALLS | #12, SYSSQIOW | |
| | 5B | | 50 | D0 | 00575 | MOVL | R0, STATUS | |
| | OC | | 5B | E9 | 00578 | BLBC | STATUS, 72\$ | 1926 |
| | 5B | F8 | AD | 3C | 0057B | MOVZWL | LOCAL_IOSB, STATUS | |
| | 05 | | 5B | E9 | 0057F | BLBC | STATUS, 72\$ | 1927 |
| | 5B | 00C4 | CE | D0 | 00582 | MOVL | FILE_FIB+52, STATUS | |
| | | 10 | AE | B5 | 00587 | TSTW | LOCAL_CHANNEL | 1928 |
| | | | OB | 12 | 0058A | BNEQ | 73\$ | |
| | 7E | 20 | AE | 3C | 0058C | MOVZWL | IO_CHANNEL, -(SP) | |
| 00000000G | 00 | | 01 | FB | 00590 | CALLS | #1, SYSSDASSGN | |
| | | 2C | AE | 9F | 00597 | PUSHAB | ACP_ATR_LIST | 1930 |
| 18 | AE | 08 | AE | D0 | 0059A | MOVL | 8(SP), 24(SP) | |
| | | 18 | AE | 9F | 0059F | PUSHAB | 24(SP) | |
| 00000000G | 00 | | 02 | FB | 005A2 | CALLS | #2, LIB\$FREE_VM | |
| | 06 | | 5B | E9 | 005A9 | BLBC | STATUS, 74\$ | 1931 |
| | 03 | | 50 | E8 | 005AC | BLBS | STATUS2, 74\$ | |
| | 5B | | 50 | D0 | 005AF | MOVL | STATUS2, STATUS | |
| | | 28 | AE | D5 | 005B2 | TSTL | LOCAL_LOCKID | 1935 |
| | | | 33 | 13 | 005B5 | BEQL | 77\$ | |
| | | | 7E | 7C | 005B7 | CLRQ | -(SP) | |
| | | | 7E | D4 | 005B9 | CLRL | -(SP) | |
| | | 34 | AE | DD | 005BB | PUSHL | LOCAL_LOCKID | |
| 00000000G | 00 | | 04 | FB | 005BE | CALLS | #4, SYSSDEQ | |
| | | | 20 | 11 | 005C5 | BRB | 76\$ | |
| 30 | AE | | 03 | D0 | 005C7 | MOVL | #3, CMK_ARG_LIST | 1943 |
| 34 | AE | | 57 | D0 | 005CB | MOVL | ITEM_COUNT, CMK_ARG_LIST+4 | 1944 |
| 38 | AE | | 59 | D0 | 005CF | MOVL | R9, CMK_ARG_LIST+8 | 1945 |
| 3C | AE | | 5A | 9A | 005D3 | MOVZBL | SHARE, CMK_ARG_LIST+12 | 1946 |
| | | 30 | AE | 9F | 005D7 | PUSHAB | CMK_ARG_LIST | 1948 |
| | | 00000000V | EF | 9F | 005DA | PUSHAB | ACL_DISPATCH | |
| 00000000G | 00 | | 02 | FB | 005E0 | CALLS | #2, SYSSCMKRN | |
| | 5B | | 50 | D0 | 005E7 | MOVL | R0, STATUS | |
| | | 10 | AE | B5 | 005EA | TSTW | LOCAL_CHANNEL | 1953 |
| | | | OB | 12 | 005ED | BNEQ | 78\$ | |
| | 7E | 20 | AE | 3C | 005EF | MOVZWL | IO_CHANNEL, -(SP) | |
| 00000000G | 00 | | 01 | FB | 005F3 | CALLS | #1, SYSSDASSGN | |
| | 20 | 1C | AE | E9 | 005FA | BLBC | 28(SP), 81\$ | 1957 |
| 1C | BC | 04 | 00 | 0D | 005FE | PROBEW | #0, #4, @CONTEXT | 1958 |
| | | | 16 | 13 | 00603 | BEQL | 80\$ | |
| | 08 | 18 | AE | E9 | 00605 | BLBC | 24(SP), 79\$ | 1961 |
| 1C | BC | 00C0 | CE | D0 | 00609 | MOVL | FILE_FIB+48, @CONTEXT | 1962 |
| | | | 0D | 11 | 0060F | BRB | 81\$ | |
| 1C | BC | 00000000' | EF | D0 | 00611 | MOVL | ACL_CONTEXT, @CONTEXT | 1963 |
| | | | 03 | 11 | 00619 | BRB | 81\$ | 1958 |
| | 5B | | OC | D0 | 0061B | MOVL | #12, STATUS | 1965 |
| | 50 | | 5B | D0 | 0061E | MOVL | STATUS, R0 | 1967 |
| | | | 04 | 00621 | RET | | | 1969 |

; Routine Size: 1570 bytes, Routine Base: \$CODE\$ + 1318

```
1976 1970 1 $SBTTL 'GET_PARENT_LOCK - get parent for ACL locks'
1977 1971 1 ROUTINE GET_PARENT_LOCK =
1978 1972 1
1979 1973 1 ++
1980 1974 1
1981 1975 1 FUNCTIONAL DESCRIPTION:
1982 1976 1
1983 1977 1 This routine takes out a null lock on the system-wide ACL lock
1984 1978 1 parent name. This lock is used as a parent for user mode ACL locks.
1985 1979 1 It must be taken out in kernel mode, since some ACL locks are
1986 1980 1 taken out in kernel mode.
1987 1981 1 numeric value. If the name does not exist, an error is returned.
1988 1982 1
1989 1983 1 CALLING SEQUENCE:
1990 1984 1 GET_PARENT_LOCK ()
1991 1985 1
1992 1986 1 INPUT PARAMETERS:
1993 1987 1 none
1994 1988 1
1995 1989 1 IMPLICIT INPUTS:
1996 1990 1 none
1997 1991 1
1998 1992 1 OUTPUT PARAMETERS:
1999 1993 1 none
2000 1994 1
2001 1995 1 IMPLICIT OUTPUTS:
2002 1996 1 PARENT_ID: set to lock ID of parent lock
2003 1997 1
2004 1998 1 ROUTINE VALUE:
2005 1999 1 Status of $END call
2006 2000 1
2007 2001 1 SIDE EFFECTS:
2008 2002 1 none
2009 2003 1
2010 2004 1 --
2011 2005 1
2012 2006 2 BEGIN
2013 2007 2
2014 2008 2 LOCAL
2015 2009 2 STATUS, ! system status return
2016 2010 2 LOCAL_IOSB : VECTOR [4, WORD]; ! lock status block
2017 2011 2
2018 2012 2
2019 2013 2 STATUS = $END (LKMODE = LCK$K_NLMODE,
2020 2014 2 LKSB = LOCAL_IOSB,
2021 2015 2 RESNAM = .LOCK_PREF[x[0],
2022 2016 2 FLAGS = LCK$M_NOQUEUE OR
2023 2017 2 LCK$M_SYNCSTS OR
2024 2018 2 LCK$M_SYSTEM,
2025 2019 2 ACMODE = PSL$C_USER);
2026 2020 2 IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
2027 2021 2 IF NOT .STATUS THEN RETURN .STATUS;
2028 2022 2 PARENT_ID = .(LOCAL_IOSB[2])<0,32>;
2029 2023 2
2030 2024 2 1
2031 2025 2 END; ! End of routine GET_PARENT_LOCK
```

```
0000 00000 GET_PARENT_LOCK:
      5E      08 C2 00002      .WORD      Save nothing      1971
      7E      03 7D 00005      SUBL2      #8, SP
      00000000' EF DD 0000C      MOVQ      #3, -(SP)      2019
      20      7E 7C 00008      CLRQ      -(SP)
      00000000G 00      7E 7C 0000A      CLRQ      -(SP)
      11      1C DD 00012      PUSHL     LOCK_PREFIX
      50      AE 9F 00014      PUSHL     #28
      08      7E 7C 00017      PUSHAB    LOCAL_IOSB
      50      0B FB 00019      CLRQ      -(SP)-
      0B      50 E9 00020      CALLS     #11, SYS$ENQ      2020
      EF      6E 3C 00023      BLBC      STATUS, 1$
      50      50 E9 00026      MOVZWL    LOCAL_IOSB, STATUS
      00000000' 04      AE D0 00029      BLBC      STATUS, 1$      2021
      50      01 D0 00031      MOVL      LOCAL_IOSB+4, PARENT_ID      2022
      04      04 00034 1$:      MOVL      #1, R0      2025
      RET
```

; Routine Size: 53 bytes, Routine Base: \$CODE\$ + 193A

SET_ID - TPARSE action routine

```
2033 2026 1 %SBTTL 'SET_ID - TPARSE action routine'
2034 2027 1 ROUTINE SET_ID =
2035 2028 1
2036 2029 1 ++
2037 2030 1
2038 2031 1 FUNCTIONAL DESCRIPTION:
2039 2032 1
2040 2033 1 This routine tries to convert an identifier to its corresponding
2041 2034 1 numeric value. If the name does not exist, an error is returned.
2042 2035 1
2043 2036 1 CALLING SEQUENCE:
2044 2037 1 SET_ID ()
2045 2038 1
2046 2039 1 INPUT PARAMETERS:
2047 2040 1 none
2048 2041 1
2049 2042 1 IMPLICIT INPUTS:
2050 2043 1 ACE_BUFFER: address of the binary ACE storage
2051 2044 1 ACE_INDEX: index into the ACE key area
2052 2045 1
2053 2046 1 OUTPUT PARAMETERS:
2054 2047 1 none
2055 2048 1
2056 2049 1 IMPLICIT OUTPUTS:
2057 2050 1 ACE_INDEX: index into the ACE key area
2058 2051 1
2059 2052 1 ROUTINE VALUE:
2060 2053 1 $$$_NORMAL if the ID name exists
2061 2054 1 $$$_NOSUCHID if it does not
2062 2055 1
2063 2056 1 SIDE EFFECTS:
2064 2057 1 The identifier name is converted to its corresponding value. That
2065 2058 1 value is then placed in the ACE key area. The index is then updated
2066 2059 1 to point to the next available key storage area.
2067 2060 1
2068 2061 1 --
2069 2062 1
2070 2063 2 BEGIN
2071 2064 2
2072 2065 2 LOCAL
2073 2066 2 UIC_POINTER : REF $BBLOCK; ! Pointer to UIC entry
2074 2067 2
2075 2068 2 ! Save the identifier, and note the type.
2076 2069 2
2077 2070 2 VECTOR [ACE_BUFFER[ACESL_KEY], .ACE_INDEX] = .IDENTIFIER;
2078 2071 2 ACE_INDEX = .ACE_INDEX + 1;
2079 2072 2 IF .IDENTIFIER[UIC$V FORMAT] EQL UIC$K_UIC_FORMAT
2080 2073 2 THEN UIC_COUNT = .UIC_COUNT + 1
2081 2074 2 ELSE ID_COUNT = .ID_COUNT + 1;
2082 2075 2
2083 2076 2 RETURN 1;
2084 2077 2
2085 2078 1 END; ! End of routine SET_ID
```

| | | | | | | | |
|-----------|----------|----|----|--------|---------------|------------------------------|--------|
| 52 | 00000000 | EF | 9E | 000002 | SET_ID: .WORD | Save R2 | : 2027 |
| 50 | | 62 | D0 | 000009 | MOVAB | ACE_INDEX, R2 | : 2070 |
| FE08 C240 | 14 | A2 | D0 | 00000C | MOVL | ACE_INDEX, R0 | : 2071 |
| CO 8F | 17 | 62 | D6 | 000013 | MOVL | IDENTIFIER, ACE_BUFFER+8[R0] | : 2072 |
| | 10 | A2 | 93 | 000015 | INCL | ACE_INDEX | : 2073 |
| | 20 | 05 | 12 | 00001A | BITB | IDENTIFIER+3, #192 | : 2074 |
| | | A2 | D6 | 00001C | BNEQ | 1\$ | : 2076 |
| | | 03 | 11 | 00001F | INCL | UIC_COUNT | : 2078 |
| 50 | | A2 | D6 | 000021 | BRB | 2\$ | |
| | | 01 | D0 | 000024 | INCL | ID_COUNT | |
| | | 04 | D0 | 000027 | MOVL | #1, R0 | |
| | | | | | RET | | |

; Routine Size: 40 bytes, Routine Base: \$CODE\$ + 196F

```
2087 2079 1 %SBTTL 'SET_ACCESS_BIT - TPARSE action routine'
2088 2080 1 ROUTINE SET_ACCESS_BIT (ARG1, ARG2, ARG3, SIZE, BUFFER) =
2089 2081 1
2090 2082 1 ++
2091 2083 1
2092 2084 1 FUNCTIONAL DESCRIPTION:
2093 2085 1
2094 2086 1     This routine checks the current token to determine whether or not
2095 2087 1     it is an access bit name.  If it is, the appropriate bit is set
2096 2088 1     in ACE_RIGHTS.  If it is not, an error is returned.
2097 2089 1
2098 2090 1 CALLING SEQUENCE:
2099 2091 1     SET_ACCESS_BIT (ARG1, ARG2, ARG3, ARG4, ARG5)
2100 2092 1
2101 2093 1 INPUT PARAMETERS:
2102 2094 1     ARG1-ARG3: TPARSE block arguments not used
2103 2095 1     ARG4: size of the current token
2104 2096 1     ARG5: address of the current token text
2105 2097 1
2106 2098 1 IMPLICIT INPUTS:
2107 2099 1     none
2108 2100 1
2109 2101 1 OUTPUT PARAMETERS:
2110 2102 1     none
2111 2103 1
2112 2104 1 IMPLICIT OUTPUTS:
2113 2105 1     ACE_RIGHTS
2114 2106 1
2115 2107 1 ROUTINE VALUE:
2116 2108 1     1 if bit name was defined
2117 2109 1     0 otherwise
2118 2110 1
2119 2111 1 SIDE EFFECTS:
2120 2112 1     The appropriate bit is set in ACE_RIGHTS.
2121 2113 1
2122 2114 1 --
2123 2115 1
2124 2116 2 BEGIN
2125 2117 2
2126 2118 2 LOCAL
2127 2119 2     BIT_POSITION,          ! Bit index
2128 2120 2     BIT_NAME_DESC : REF $BBLOCK; ! Bit name descriptor
2129 2121 2
2130 2122 2 ! Note that, initially, no match was found.
2131 2123 2
2132 2124 2 BIT_POSITION = -1;
2133 2125 2
2134 2126 2 ! Now scan the bit name table to see if the specified definition exists.
2135 2127 2
2136 2128 2 INCR J FROM 0 TO 31
2137 2129 2 DO
2138 2130 2     BEGIN
2139 2131 2     IF .BIT_NAME_TABLE NEQA 0
2140 2132 2     THEN
2141 2133 2         BEGIN
2142 2134 2         IF PROBER (%REF (0), %REF (DSC$C S BLN), BIT_NAME_TABLE[J, 0, 0, 0, 0])
2143 2135 2         THEN BIT_NAME_DESC = BIT_NAME_TABLE[J, 0, 0, 0, 0]
```

[illegible]

| PC | Op | Op2 | Op3 | Op4 | Op5 | Op6 | Op7 | Op8 | Op9 | Op10 | Op11 | Op12 | Op13 | Op14 | Op15 | Op16 | Op17 | Op18 | Op19 | Op20 | Op21 | Op22 | Op23 | Op24 | Op25 | Op26 | Op27 | Op28 | Op29 | Op30 | Op31 | Op32 | Op33 | Op34 | Op35 | Op36 | Op37 | Op38 | Op39 | Op40 | Op41 | Op42 | Op43 | Op44 | Op45 | Op46 | Op47 | Op48 | Op49 | Op50 | Op51 | Op52 | Op53 | Op54 | Op55 | Op56 | Op57 | Op58 | Op59 | Op60 | Op61 | Op62 | Op63 | Op64 | Op65 | Op66 | Op67 | Op68 | Op69 | Op70 | Op71 | Op72 | Op73 | Op74 | Op75 | Op76 | Op77 | Op78 | Op79 | Op80 | Op81 | Op82 | Op83 | Op84 | Op85 | Op86 | Op87 | Op88 | Op89 | Op90 | Op91 | Op92 | Op93 | Op94 | Op95 | Op96 | Op97 | Op98 | Op99 | Op100 | Op101 | Op102 | Op103 | Op104 | Op105 | Op106 | Op107 | Op108 | Op109 | Op110 | Op111 | Op112 | Op113 | Op114 | Op115 | Op116 | Op117 | Op118 | Op119 | Op120 | Op121 | Op122 | Op123 | Op124 | Op125 | Op126 | Op127 | Op128 | Op129 | Op130 | Op131 | Op132 | Op133 | Op134 | Op135 | Op136 | Op137 | Op138 | Op139 | Op140 | Op141 | Op142 | Op143 | Op144 | Op145 | Op146 | Op147 | Op148 | Op149 | Op150 | Op151 | Op152 | Op153 | Op154 | Op155 | Op156 | Op157 | Op158 | Op159 | Op160 | Op161 | Op162 | Op163 | Op164 | Op165 | Op166 | Op167 | Op168 | Op169 | Op170 | Op171 | Op172 | Op173 | Op174 | Op175 | Op176 | Op177 | Op178 | Op179 | Op180 | Op181 | Op182 | Op183 | Op184 | Op185 | Op186 | Op187 | Op188 | Op189 | Op190 | Op191 | Op192 | Op193 | Op194 | Op195 | Op196 | Op197 | Op198 | Op199 | Op200 | Op201 | Op202 | Op203 | Op204 | Op205 | Op206 | Op207 | Op208 | Op209 | Op210 | Op211 | Op212 | Op213 | Op214 | Op215 | Op216 | Op217 | Op218 | Op219 | Op220 | Op221 | Op222 | Op223 | Op224 | Op225 | Op226 | Op227 | Op228 | Op229 | Op230 | Op231 | Op232 | Op233 | Op234 | Op235 | Op236 | Op237 | Op238 | Op239 | Op240 | Op241 | Op242 | Op243 | Op244 | Op245 | Op246 | Op247 | Op248 | Op249 | Op250 | Op251 | Op252 | Op253 | Op254 | Op255 | Op256 | Op257 | Op258 | Op259 | Op260 | Op261 | Op262 | Op263 | Op264 | Op265 | Op266 | Op267 | Op268 | Op269 | Op270 | Op271 | Op272 | Op273 | Op274 | Op275 | Op276 | Op277 | Op278 | Op279 | Op280 | Op281 | Op282 | Op283 | Op284 | Op285 | Op286 | Op287 | Op288 | Op289 | Op290 | Op291 | Op292 | Op293 | Op294 | Op295 | Op296 | Op297 | Op298 | Op299 | Op300 | Op301 | Op302 | Op303 | Op304 | Op305 | Op306 | Op307 | Op308 | Op309 | Op310 | Op311 | Op312 | Op313 | Op314 | Op315 | Op316 | Op317 | Op318 | Op319 | Op320 | Op321 | Op322 | Op323 | Op324 | Op325 | Op326 | Op327 | Op328 | Op329 | Op330 | Op331 | Op332 | Op333 | Op334 | Op335 | Op336 | Op337 | Op338 | Op339 | Op340 | Op341 | Op342 | Op343 | Op344 | Op345 | Op346 | Op347 | Op348 | Op349 | Op350 | Op351 | Op352 | Op353 | Op354 | Op355 | Op356 | Op357 | Op358 | Op359 | Op360 | Op361 | Op362 | Op363 | Op364 | Op365 | Op366 | Op367 | Op368 | Op369 | Op370 | Op371 | Op372 | Op373 | Op374 | Op375 | Op376 | Op377 | Op378 | Op379 | Op380 | Op381 | Op382 | Op383 | Op384 | Op385 | Op386 | Op387 | Op388 | Op389 | Op390 | Op391 | Op392 | Op393 | Op394 | Op395 | Op396 | Op397 | Op398 | Op399 | Op400 | Op401 | Op402 | Op403 | Op404 | Op405 | Op406 | Op407 | Op408 | Op409 | Op410 | Op411 | Op412 | Op413 | Op414 | Op415 | Op416 | Op417 | Op418 | Op419 |
|----|----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|----|----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|

SYSACLSRV
V04-000

SET_ACCESS_BIT - TPARSE action routine

L 3
16-Sep-1984 01:51:51
14-Sep-1984 12:40:53

VAX-11 BLISS-32 V4.0-742
[LOADSS.SRC]SYSACLSRV.B32;1

Page 93
(9)

00 00000000' EF
50

56 E2 00063
01 D0 0006B 7\$:
04 0006E
50 D4 0006F 8\$:
04 00071

BBSS
MOVL
RET
CLRL
RET

BIT_POSITION, ACE_RIGHTS, 7\$
#1, -R0
R0

: 2152
: 2154
: 2156
:

; Routine Size: 114 bytes, Routine Base: \$CODE\$ + 1997

```
2166 2157 1 %SBTTL 'GET UCB ACL - get UCB ACL queue head address'
2167 2158 1 ROUTINE GET_UCB_ACL (CHANNEL) =
2168 2159 1
2169 2160 1 ++
2170 2161 1
2171 2162 1 FUNCTIONAL DESCRIPTION:
2172 2163 1
2173 2164 1 This routine locates the ACL queue head for a device, given a
2174 2165 1 channel number.
2175 2166 1
2176 2167 1 CALLING SEQUENCE:
2177 2168 1 GET_UCB_ACL (ARG1)
2178 2169 1
2179 2170 1 INPUT PARAMETERS:
2180 2171 1 ARG1: channel assigned to the device
2181 2172 1
2182 2173 1 IMPLICIT INPUTS:
2183 2174 1 none
2184 2175 1
2185 2176 1 OUTPUT PARAMETERS:
2186 2177 1 none
2187 2178 1
2188 2179 1 IMPLICIT OUTPUTS:
2189 2180 1 none
2190 2181 1
2191 2182 1 ROUTINE VALUE:
2192 2183 1 $$$_NORMAL if ACLs are allowed
2193 2184 1 $$$_NOACLSUPPORT if ACLs are not allowed
2194 2185 1
2195 2186 1 SIDE EFFECTS:
2196 2187 1 none
2197 2188 1
2198 2189 1 --
2199 2190 1
2200 2191 2 BEGIN
2201 2192 2
2202 2193 2 MAP
2203 2194 2 CHANNEL : WORD;
2204 2195 2
2205 2196 2 LOCAL
2206 2197 2 STATUS, ! Local routine return status
2207 2198 2 DEVICE_UCB : REF $BBLOCK, ! Device UCB address
2208 2199 2 DEVICE_ORB : REF $BBLOCK, ! Device ORB address
2209 2200 2 CHANNEL_BLOCK : REF $BBLOCK; ! Device CCB address
2210 2201 2
2211 2202 2 STATUS = IOC$VERIFYCHAN (.CHANNEL; CHANNEL_BLOCK);
2212 2203 2 IF NOT .STATUS THEN RETURN .STATUS;
2213 2204 2 DEVICE_UCB = .CHANNEL_BLOCK[CCB$L_UCB];
2214 2205 2 DEVICE_ORB = .DEVICE_UCB[UCB$L_ORB];
2215 2206 2
2216 2207 2 ! If no ACLs are allowed, return an error.
2217 2208 2
2218 2209 2 IF .DEVICE_ORB[ORB$V_NOACL] THEN RETURN $$$_NOACLSUPPORT;
2219 2210 2
2220 2211 2 ! If the device is unowned, and the protection is all access to everybody,
2221 2212 2 ! and there is no ACL present, require SYSPRV to change the ACL.
2222 2213 2
```

```
2223 2 IF .DEVICE_ORB[ORBSL_OWNER] EQL 0
2224 AND (IF .DEVICE_ORB[ORBSV_ACL_QUEUE]
2225 THEN .DEVICE_ORB[ORBSL_ACLFL] EQLA DEVICE_ORB[ORBSL_ACLFL]
2226 ELSE 1)
2227 AND NOT .SBBLOCK [CTL$GL_PCB[PCBSQ_PRIV], PRVSV_SYSPRV]
2228 THEN RETURN SSS_NOPRV;
2229
2230 ! If the ACL queue head is uninitialized, initialize it now.
2231
2232 IF NOT .DEVICE_ORB[ORBSV_ACL_QUEUE] THEN ACL_INIT_QUEUE (.DEVICE_ORB);
2233
2234 ! Set up the address of the ACL queue head.
2235
2236 ACL_QUEUE_HEAD = SBBLOCK [.DEVICE_UCB[UCBSL_ORB], ORBSL_ACLFL];
2237
2238 RETURN SSS_NORMAL;
2239
2240 1 END;                                     ! End of routine GET_UCB_ACL
```

| | | | | 000C 00000 GET_UCB_ACL: | | | | |
|-----------|------|----|-----------|-------------------------|----------|--------|-------------------------------------|------|
| | | 50 | 04 | AC | 3C 00002 | .WORD | Save R2,R3 | 2158 |
| | | | 00000000G | 00 | 16 00006 | MOVZWL | CHANNEL, R0 | 2202 |
| | | 50 | | 50 | E9 0000C | JSB | IOCSVERIFYCHAN | |
| | | 52 | | 61 | D0 0000F | BLBC | STATUS, 5\$ | 2203 |
| | | 50 | 1C | A2 | D0 00012 | MOVL | (CHANNEL_BLOCK), DEVICE_UCB | 2204 |
| 06 | 0B | A0 | | 03 | E1 00016 | MOVL | 28(DEVICE_UCB), DEVICE_ORB | 2205 |
| | | 50 | 22BC | 8F | 3C 0001B | BBC | #3, 11(DEVICE_ORB), 1\$ | 2209 |
| | | | | | 04 00020 | MOVZWL | #8892, R0 | |
| | | | | 60 | D5 00021 | RET | | |
| | | | | 20 | 12 00023 | TSTL | (DEVICE_ORB) | 2214 |
| 0A | 0B | A0 | | 01 | E1 00025 | BNEQ | 3\$ | |
| | | 51 | 28 | A0 | 9E 0002A | BBC | #1, 11(DEVICE_ORB), 2\$ | 2215 |
| | | 51 | 28 | A0 | D1 0002E | MOVAB | 40(DEVICE_ORB), R1 | 2216 |
| | | | | 11 | 12 00032 | CMP | 40(DEVICE_ORB), R1 | |
| | | 51 | 00000000G | 00 | D0 00034 | BNEQ | 3\$ | |
| 04 | 0087 | C1 | | 04 | E0 0003B | MOVL | CTL\$GL_PCB, R1 | 2218 |
| | | 50 | | 24 | D0 00041 | BBS | #4, 135(R1), 3\$ | |
| | | | | | 04 00044 | MOVL | #36, R0 | 2219 |
| | | | | | | RET | | |
| 09 | 0B | A0 | | 01 | E0 00045 | BBS | #1, 11(DEVICE_ORB), 4\$ | 2223 |
| | | | | 50 | DD 0004A | PUSHL | DEVICE_ORB | |
| | | 00 | | 01 | FB 0004C | CALLS | #1, ACL_INIT_QUEUE | |
| 00000000' | EF | A2 | | 28 | C1 00053 | ADDL3 | #40, 28(DEVICE_UCB), ACL_QUEUE_HEAD | 2227 |
| | | 50 | | 01 | D0 0005C | MOVL | #1, R0 | 2229 |
| | | | | 04 | 0005F | RET | | 2231 |

; Routine Size: 96 bytes, Routine Base: \$CODE\$ + 1A09

```
2242 1 %SBTTL 'GET_JBC_ACL - get Job Controller queue ACL queue head address'
2243 1 ROUTINE GET_JBC_ACL (QUEUE_NAME) =
2244 1
2245 1 ++
2246 1
2247 1 FUNCTIONAL DESCRIPTION:
2248 1
2249 1 This routine locates the ACL queue head for a Job Controller
2250 1 queue.
2251 1
2252 1 CALLING SEQUENCE:
2253 1 GET_JBC_ACL (ARG1)
2254 1
2255 1 INPUT PARAMETERS:
2256 1 ARG1: address of the queue name descriptor
2257 1
2258 1 IMPLICIT INPUTS:
2259 1 none
2260 1
2261 1 OUTPUT PARAMETERS:
2262 1 none
2263 1
2264 1 IMPLICIT OUTPUTS:
2265 1 none
2266 1
2267 1 ROUTINE VALUE:
2268 1 address of the ACL queue head or 0 if an error has occurred
2269 1
2270 1 SIDE EFFECTS:
2271 1 none
2272 1
2273 1 --
2274 1
2275 2 BEGIN
2276 2
2277 2 MAP
2278 2 QUEUE_NAME : REF $BLOCK;
2279 2
2280 2 RETURN $$$_BADPARAM;
2281 2
2282 1 END; ! End of routine GET_JBC_ACL
```

0000 00000 GET_JBC_ACL:

50

14 D0 00002
04 00005.WORD
MOVL
RETSave nothing
#20, R0: 2233
: 2270
: 2272

; Routine Size: 6 bytes, Routine Base: \$CODE\$ + 1A69


```
2284 2273 1 %SBTTL 'GET_CEB_ACL - get common event block ACL queue head address'
2285 2274 1 ROUTINE GET_CEB_ACL (CLOSTER_NAME) =
2286 2275 1
2287 2276 1 ++
2288 2277 1
2289 2278 1 FUNCTIONAL DESCRIPTION:
2290 2279 1
2291 2280 1 This routine locates the ACL queue head for a common event block.
2292 2281 1
2293 2282 1 CALLING SEQUENCE:
2294 2283 1 GET_CEB_ACL (ARG1)
2295 2284 1
2296 2285 1 INPUT PARAMETERS:
2297 2286 1 ARG1: address of the cluster name descriptor
2298 2287 1
2299 2288 1 IMPLICIT INPUTS:
2300 2289 1 none
2301 2290 1
2302 2291 1 OUTPUT PARAMETERS:
2303 2292 1 none
2304 2293 1
2305 2294 1 IMPLICIT OUTPUTS:
2306 2295 1 none
2307 2296 1
2308 2297 1 ROUTINE VALUE:
2309 2298 1 address of the ACL queue head or 0 if an error has occurred
2310 2299 1
2311 2300 1 SIDE EFFECTS:
2312 2301 1 none
2313 2302 1
2314 2303 1 --
2315 2304 1
2316 2305 2 BEGIN
2317 2306 2
2318 2307 2 MAP
2319 2308 2 CLUSTER_NAME : REF $BBLOCK;
2320 2309 2
2321 2310 2 RETURN SS$_BADPARAM;
2322 2311 2
2323 2312 1 END; ! End of routine GET_CEB_ACL
```

0000 00000 GET_CEB_ACL:

50

14 D0 00002
04 00005WORD
MOVL
RETSave nothing
#20, R0: 2274
: 2310
: 2312

; Routine Size: 6 bytes, Routine Base: \$CODE\$ + 1A6F

GET_LNT_ACL - get logical name table ACL queue

```
2325 1 XSBTTL 'GET_LNT_ACL - get logical name table ACL queue head address'
2326 1 ROUTINE GET_LNT_ACL (TABLE_NAME) =
2327 1
2328 1 ++
2329 1
2330 1 FUNCTIONAL DESCRIPTION:
2331 1
2332 1     This routine locates the ACL queue head for aLogical name
2333 1     table.
2334 1
2335 1 CALLING SEQUENCE:
2336 1     GET_LNT_ACL (ARG1)
2337 1
2338 1 INPUT PARAMETERS:
2339 1     ARG1: address of the table name descriptor
2340 1
2341 1 IMPLICIT INPUTS:
2342 1     none
2343 1
2344 1 OUTPUT PARAMETERS:
2345 1     none
2346 1
2347 1 IMPLICIT OUTPUTS:
2348 1     none
2349 1
2350 1 ROUTINE VALUE:
2351 1     address of the ACL queue head or 0 if an error has occurred
2352 1
2353 1 SIDE EFFECTS:
2354 1     none
2355 1
2356 1 --
2357 1
2358 2 BEGIN
2359 2
2360 2 RETURN SS$_BADPARAM;
2361 2
2362 1 END;
```

! End of routine GET_LNT_ACL

0000 00000 GET_LNT_ACL:

50

14 D0 00002
04 00005.WORD
MOVL
RETSave nothing
#20, R0: 2314
: 2348
: 2350

; Routine Size: 6 bytes, Routine Base: \$CODE\$ + 1A75

```
2364 2351 1 XSBTTL 'GET_PCB_ACL - get process ACL queue head address'
2365 2352 1 ROUTINE GET_PCB_ACL (PROCESS_NAME) =
2366 2353 1
2367 2354 1 ++
2368 2355 1
2369 2356 1 FUNCTIONAL DESCRIPTION:
2370 2357 1
2371 2358 1 This routine locates the ACL queue head for a process.
2372 2359 1
2373 2360 1 CALLING SEQUENCE:
2374 2361 1 GET_PCB_ACL (ARG1)
2375 2362 1
2376 2363 1 INPUT PARAMETERS:
2377 2364 1 ARG1: address of the process name descriptor
2378 2365 1
2379 2366 1 IMPLICIT INPUTS:
2380 2367 1 none
2381 2368 1
2382 2369 1 OUTPUT PARAMETERS:
2383 2370 1 none
2384 2371 1
2385 2372 1 IMPLICIT OUTPUTS:
2386 2373 1 none
2387 2374 1
2388 2375 1 ROUTINE VALUE:
2389 2376 1 address of the ACL queue head or 0 if an error has occurred
2390 2377 1
2391 2378 1 SIDE EFFECTS:
2392 2379 1 none
2393 2380 1
2394 2381 1 --
2395 2382 1
2396 2383 2 BEGIN
2397 2384 2
2398 2385 2 RETURN SS$_BADPARAM;
2399 2386 2
2400 2387 1 END; ! End of routine GET_PCB_ACL
```

0000 00000 GET_PCB_ACL:

50

14 D0 00002
04 00005.WORD
MOVL
RETSave nothing
#20, R0: 2352
: 2385
: 2387

; Routine Size: 6 bytes, Routine Base: \$CODE\$ + 1A7B

```
2402 2388 1 XSBTTL 'GET_GBL_ACL - get global section ACL queue head address'
2403 2389 1 ROUTINE GET_GBL_ACL (SECTION_NAME) =
2404 2390 1
2405 2391 1 ++
2406 2392 1
2407 2393 1 FUNCTIONAL DESCRIPTION:
2408 2394 1
2409 2395 1 This routine locates the ACL queue head for a global section, given
2410 2396 1 a section name.
2411 2397 1
2412 2398 1 CALLING SEQUENCE:
2413 2399 1 GET_GBL_ACL (ARG1)
2414 2400 1
2415 2401 1 INPUT PARAMETERS:
2416 2402 1 ARG1: address of the section name descriptor
2417 2403 1
2418 2404 1 IMPLICIT INPUTS:
2419 2405 1 none
2420 2406 1
2421 2407 1 OUTPUT PARAMETERS:
2422 2408 1 none
2423 2409 1
2424 2410 1 IMPLICIT OUTPUTS:
2425 2411 1 none
2426 2412 1
2427 2413 1 ROUTINE VALUE:
2428 2414 1 address of the ACL queue head or 0 if an error has occurred
2429 2415 1
2430 2416 1 SIDE EFFECTS:
2431 2417 1 none
2432 2418 1
2433 2419 1 --
2434 2420 1
2435 2421 2 BEGIN
2436 2422 2
2437 2423 2 RETURN SSS_BADPARAM;
2438 2424 2
2439 2425 1 END; ! End of routine GET_GBL_ACL
```

```
0000 00000 GET_GBL_ACL:
50 14 D0 00002 .WORD Save nothing
04 00005 MOVL #20, R0
RET
: 2389
: 2423
: 2425

; Routine Size: 6 bytes, Routine Base: $CODE$ + 1A81
```



```
2441 2426 1 XSBTTL 'ACL_DISPATCH - main ACL function dispatcher'
2442 2427 1 ROUTINE ACL_DISPATCH (ITEM_COUNT, ITEM_LIST, SHARE) =
2443 2428 1
2444 2429 1 ++
2445 2430 1
2446 2431 1 FUNCTIONAL DESCRIPTION:
2447 2432 1
2448 2433 1 This routine is called to perform the appropriate ACL operations.
2449 2434 1 The code is checked for validity and, when necessary, the buffer
2450 2435 1 is probed for the desired access.
2451 2436 1
2452 2437 1 CALLING SEQUENCE:
2453 2438 1 ACL_DISPATCH (ARG1, ARG2, ARG3)
2454 2439 1
2455 2440 1 INPUT PARAMETERS:
2456 2441 1 ARG1: count of items to process
2457 2442 1 ARG2: address of the item list
2458 2443 1 ARG3: 1 if the operation is only reading the ACL
2459 2444 1 0 if the ACL is being modified
2460 2445 1
2461 2446 1 IMPLICIT INPUTS:
2462 2447 1 ACL_CONTEXT: previous ACL context
2463 2448 1
2464 2449 1 OUTPUT PARAMETERS:
2465 2450 1 NONE
2466 2451 1
2467 2452 1 IMPLICIT OUTPUTS:
2468 2453 1 ACL_CONTEXT: new ACL context
2469 2454 1
2470 2455 1 ROUTINE VALUE:
2471 2456 1 1
2472 2457 1
2473 2458 1 SIDE EFFECTS:
2474 2459 1 The appropriate action routine is called. Possible ACL modification
2475 2460 1 may result.
2476 2461 1
2477 2462 1 --
2478 2463 1
2479 2464 2 BEGIN
2480 2465 2
2481 2466 2 MAP
2482 2467 2 ITEM_LIST : REF BLOCKVECTOR [, ITEMSS_ITEM, BYTE];
2483 2468 2
2484 2469 2 ! Cells defined to tie off references made in the module ALLOCB obtained from
2485 2470 2 ! the XQP.
2486 2471 2
2487 2472 2 GLOBAL LITERAL
2488 2473 2 CONTEXT_SAVE = 0;
2489 2474 2 CURRENT_WINDOW = 0;
2490 2475 2 IO_PACKET = 0;
2491 2476 2
2492 2477 2 LOCAL
2493 2478 2 ACL_STATUS, ! Status returned by ACL operation
2494 2479 2 STATUS, ! Routine return status
2495 2480 2 FUNCTION_CODE, ! Operation to perform
2496 2481 2 SIZE, ! Size of user buffer
2497 2482 2 BUFFER : REF $BLOCK, ! Address of user buffer
```

```
2498 2483 2      LOCAL_IOSB      : VECTOR [4, WORD],      ! Lock status block
2499 2484      LOCAL_LOCKID;      ! Local copy of the lock-id
2500 2485
2501 2486      ! Initialize local storage.
2502 2487
2503 2488      CH$FILL (0, 4*2, LOCAL_IOSB);
2504 2489      LOCAL_IOSB[0] = $$$_NORMAL;      ! Assume success
2505 2490      ACL_STATUS = STATUS = $$$_NORMAL;      ! Here also
2506 2491
2507 2492      ! Take out the mutex on the specified ACL.
2508 2493
2509 2494      IF .SHARE
2510 2495      THEN SCH$LOCKR (.ACL_QUEUE_HEAD - $BYTEOFFSET (ORBSL_ACLFL) + $BYTEOFFSET (ORBSL_ACL_MUTEX), .CTL$GL_PCB)
2511 2496      ELSE SCH$LOCKW (.ACL_QUEUE_HEAD - $BYTEOFFSET (ORBSL_ACLFL) + $BYTEOFFSET (ORBSL_ACL_MUTEX), .CTL$GL_PCB);
2512 2497
2513 2498      ! Loop over the item list, processing each item.
2514 2499
2515 2500      INCR J FROM 0 TO .ITEM_COUNT-1
2516 2501      DO
2517 2502          BEGIN
2518 2503
2519 2504          FUNCTION_CODE = .ITEM_LIST[J, ITMSW_ITMCD];
2520 2505          SIZE = .ITEM_LIST[J, ITMSW_BUFSIZ];
2521 2506          BUFFER = .ITEM_LIST[J, ITMSL_BUFADR];
2522 2507
2523 2508      ! Dispatch on the function code.
2524 2509
2525 2510      CASE .FUNCTION_CODE FROM MIN_ACL_ATR TO MAX_ACL_ATR OF
2526 2511      SET
2527 2512
2528 2513      [ACL$C_ADDACLENT]:
2529 2514      BEGIN
2530 2515      IF .SHARE
2531 2516      THEN STATUS = $$$_BADPARAM
2532 2517      ELSE IF NOT EXE$PROBER (.CALL_ACMODE, .SIZE, .BUFFER)
2533 2518      THEN STATUS = $$$_ACCVIO
2534 2519      ELSE IF .ACL_STATUS
2535 2520      THEN ACL_STATUS = ACL_ADDENTRY (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
2536 2521      END;
2537 2522
2538 2523      [ACL$C_DEACLENT]:
2539 2524      BEGIN
2540 2525      IF .SHARE
2541 2526      THEN STATUS = $$$_BADPARAM
2542 2527      ELSE IF NOT EXE$PROBER (.CALL_ACMODE, .SIZE, .BUFFER)
2543 2528      THEN STATUS = $$$_ACCVIO
2544 2529      ELSE IF .ACL_STATUS
2545 2530      THEN ACL_STATUS = ACL_DELENTY (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
2546 2531      END;
2547 2532
2548 2533      [ACL$C_MODACLENT]:
2549 2534      BEGIN
2550 2535      IF .SHARE
2551 2536      THEN STATUS = $$$_BADPARAM
2552 2537      ELSE IF NOT EXE$PROBER (.CALL_ACMODE, .SIZE, .BUFFER)
2553 2538      THEN STATUS = $$$_ACCVIO
2554 2539      ELSE IF .ACL_STATUS
```

```
.. 2555      2540      4      THEN ACL_STATUS = ACL_MODENTRY (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
.. 2556      2541      4      END;
.. 2557      2542      4
.. 2558      2543      4      [ACLSC_FNDACLENT]:
.. 2559      2544      4      BEGIN
.. 2560      2545      4      IF NOT EXESPROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
.. 2561      2546      4      THEN STATUS = SSS_ACCVIO
.. 2562      2547      4      ELSE ACL_STATUS = ACL_FINDENTRY (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER, 0);
.. 2563      2548      4      END;
.. 2564      2549      4
.. 2565      2550      4      [ACLSC_FNDACETYP]:
.. 2566      2551      4      BEGIN
.. 2567      2552      4      IF NOT EXESPROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
.. 2568      2553      4      THEN STATUS = SSS_ACCVIO
.. 2569      2554      4      ELSE ACL_STATUS = ACL_FINDTYPE (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER, 0);
.. 2570      2555      4      END;
.. 2571      2556      4
.. 2572      2557      4      [ACLSC_DELETEACL]:
.. 2573      2558      4      BEGIN
.. 2574      2559      4      IF .SHARE
.. 2575      2560      4      THEN STATUS = SSS_BADPARAM
.. 2576      2561      4      ELSE IF .ACL_STATUS
.. 2577      2562      4      THEN ACL_STATUS = ACL_DELETEACL (.ACL_QUEUE_HEAD, ACL_CONTEXT);
.. 2578      2563      4      END;
.. 2579      2564      4
.. 2580      2565      4      [ACLSC_READACL]:
.. 2581      2566      4      BEGIN
.. 2582      2567      4      IF NOT EXESPROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
.. 2583      2568      4      THEN STATUS = SSS_ACCVIO
.. 2584      2569      4      ELSE ACL_STATUS = ACL_READACL (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
.. 2585      2570      4      END;
.. 2586      2571      4
.. 2587      2572      4      [ACLSC_ACLLENGTH]:
.. 2588      2573      4      BEGIN
.. 2589      2574      4      IF NOT EXESPROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
.. 2590      2575      4      THEN STATUS = SSS_ACCVIO
.. 2591      2576      4      ELSE ACL_STATUS = ACL_ACLLENGTH (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
.. 2592      2577      4      END;
.. 2593      2578      4
.. 2594      2579      4      [ACLSC_READACE]:
.. 2595      2580      4      BEGIN
.. 2596      2581      4      IF NOT EXESPROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
.. 2597      2582      4      THEN STATUS = SSS_ACCVIO
.. 2598      2583      4      ELSE ACL_STATUS = ACL_READACE (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
.. 2599      2584      4      END;
.. 2600      2585      4
.. 2601      2586      4
.. 2602      2587      4      [ACLSC_RLOCK_ACL
.. 2603      2588      4      ACLSC_WLOCK_ACL]:
.. 2604      2589      4      BEGIN
.. 2605      2590      4      IF .SIZE LSSU 4
.. 2606      2591      4      THEN STATUS = SSS_BADPARAM
.. 2607      2592      4      ELSE IF NOT EXESPROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
.. 2608      2593      4      THEN STATUS = SSS_ACCVIO
.. 2609      2594      4      ELSE
.. 2610      2595      5      BEGIN
.. 2611      2596      5      STATUS = $ENQ (LKMODE = (IF .FUNCTION_CODE EQL ACLSC_RLOCK_ACL
```

```
2612      THEN LCK$K_CRMODE ELSE LCK$K_PWMODE),
2613      LKSB = LOCAL_IOSB,
2614      RESNAM = LOCK_RESNAM,
2615      PARID = (IF .CALL ACMODE EQL PSL$C_USER
2616              THEN .PARENT_ID
2617              ELSE 0),
2618      FLAGS = LCK$M_NOQUEUE OR
2619              LCK$M_SYNCSTS OR
2620              LCK$M_SYSTEM,
2621      ACMODE = PSL$C_USER);
2622      IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
2623      CH$COPY (4, LOCAL_IOSB[2],
2624              0,
2625              .SIZE, .BUFFER);          ! Copy lock-id
2626      END;
2627      END;
2628
2629      [ACL$C_UNLOCK_ACL]:
2630      BEGIN
2631      IF .SIZE LSSU 4
2632      THEN STATUS = SSS_BADPARAM
2633      ELSE IF NOT EXE$PROBER (.CALL ACMODE, .SIZE, .BUFFER)
2634      THEN STATUS = SSS_ACCVIO
2635      ELSE
2636      BEGIN
2637      CH$COPY (.SIZE, .BUFFER, 0, 4, LOCAL_LOCKID);
2638      STATUS = $DEQ (LKID = .LOCAL_LOCKID);
2639      END;
2640      END;
2641
2642      [INRANGE, OUTRANGE]:
2643      BEGIN
2644      STATUS = SSS_BADPARAM;
2645      END;
2646
2647      TES;
2648
2649      IF NOT .STATUS THEN EXITLOOP;
2650      END;
2651
2652      ! If an error occurred because of an access violation or an access conflict
2653      ! (trying to modify the ACL when only holding a read lock), return it. Otherwise
2654      ! return any error that may have occurred during the ACL processing.
2655
2656      IF .STATUS THEN STATUS = .ACL_STATUS;
2657
2658      ! Release the ACL mutex.
2659
2660      SCH$UNLOCK (.ACL_QUEUE_HEAD - $BYTEOFFSET (ORBSL_ACLFL) + $BYTEOFFSET (ORBSL_ACL_MUTEX), .CTL$GL_PCB);
2661
2662      RETURN .STATUS;
2663
2664      ! End of routine ACL_DISPATCH
2665      END;
```


CONTEXT_SAVE== 0
CURRENT_WINDOW== 0
IO_PACKET== 0

```
OFFC 00000 ACL_DISPATCH:
08      00      5E      10      C2      00002      .WORD      Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      : 2427
      6E      00      2C      00005      SUBL2      #16, SP
      08      AE      01      B0      0000C      MOVCS      #0, (SP), #0, #8, LOCAL_IOSB      : 2488
      59      01      D0      00010      MOVW      #1, LOCAL_IOSB      : 2489
      5B      01      D0      00013      MOVL      #1, STATUS      : 2490
      54      00000000G      00      D0      00016      MOVL      #1, ACL STATUS
      50      00000000'      24      C3      0001D      MOVL      CTL$GL PCB, R4      : 2495
      6E      0C      AC      D0      00025      SUBL3      #36, ACL_QUEUE_HEAD, R0
      08      6E      E9      00029      MOVL      SHARE, (SP)      : 2494
      00000000G      00      16      0002C      BLBC      (SP), 1$
      00000000G      00      16      00034      JSB      SCH$LOCKR      : 2495
      56      01      CE      0003A      BRB      2$
      0291      31      0003D      JSB      SCH$LOCKW      : 2496
      51      56      0C      C5      00040      MNEGL      #1, J      : 2504
      50      51      08      AC      C1      00044      BRW      46$
      5A      02      A0      3C      00049      MULL3      #12, J, R1
      50      51      08      AC      C1      0004D      ADDL3      ITEM_LIST, R1, R0
      57      60      3C      00052      MOVZWL      2(R0), FUNCTION_CODE
      50      51      08      AC      C1      00055      ADDL3      ITEM_LIST, R1, R0      : 2505
      58      04      A0      D0      0005A      MOVZWL      (R0), SIZE
      01      5A      CF      0005E      ADDL3      ITEM_LIST, R1, R0      : 2506
      0083      004F      001B      00062      4$:      MOVL      4(R0), BUFFER
      0169      011A      00EA      0006A      CASEL      FUNCTION_CODE, #1, #11      : 2510
      022F      01C8      0197      00072      .WORD      6$-4$, -
      : 7$-4$, -
      : 8$-4$, -
      : 13$-4$, -
      : 16$-4$, -
      : 18$-4$, -
      : 23$-4$, -
      : 26$-4$, -
      : 29$-4$, -
      : 32$-4$, -
      : 32$-4$, -
      : 39$-4$
      0219      31      0007A      5$:      BRW      40$      : 2630
      6E      E8      0007D      6$:      BLBS      (SP), 5$      : 2515
      50      58      D0      00080      MOVL      BUFFER, R0      : 2517
      51      57      D0      00083      MOVL      SIZE, R1
      53      00000000'      EF      D0      00086      MOVL      CALL_ACMODE, R3
      00000000G      00      16      0008D      JSB      EXE$PROBER
      65      50      E9      00093      BLBC      R0, 9$
      65      5B      E9      00096      BLBC      ACL_STATUS, 10$
      7E      57      7D      00099      MOVQ      SIZE, -(SP)
      00000000'      EF      9F      0009C      PUSHAB      ACL_CONTEXT
      00000000'      EF      DD      000A2      PUSHL      ACL_QUEUE_HEAD
      00000000G      00      04      FB      000AB      CALLS      #4, ACL_ADDENTRY
      69      11      000AF      BRB      12$
      C6      6E      E8      000B1      7$:      BLBS      (SP), 5$
      50      5B      D0      000B4      MOVL      BUFFER, R0      : 2525
      : 2527
```

| | | | | | | | |
|-----------|-----------|------|----|-------|--------|-------------------|------|
| 51 | | 57 | D0 | 000B7 | MOVL | SIZE, R1 | |
| 53 | 00000000' | EF | D0 | 000BA | MOVL | CALL ACMODE, R3 | |
| | 00000000G | 00 | 16 | 000C1 | JSB | EXESPROBER | |
| 65 | | 50 | E9 | 000C7 | BLBC | RO, 14\$ | |
| 31 | | 58 | E9 | 000CA | BLBC | ACL_STATUS, 10\$ | 2529 |
| 7E | | 57 | 7D | 000CD | MOVQ | SIZE, -(SP) | 2530 |
| | 00000000' | EF | 9F | 000D0 | PUSHAB | ACL_CONTEXT | |
| | 00000000' | EF | DD | 000D6 | PUSHL | ACL_QUEUE_HEAD | |
| 00000000G | 00 | 04 | FB | 000DC | CALLS | #4, ACL_DELENTY | |
| | | 65 | 11 | 000E3 | BRB | 15\$ | |
| 92 | | 6E | E8 | 000E5 | BLBS | (SP), 5\$ | 2535 |
| 50 | | 58 | D0 | 000E8 | MOVL | BUFFER, RO | 2537 |
| 51 | | 57 | D0 | 000EB | MOVL | SIZE, R1 | |
| 53 | 00000000' | EF | D0 | 000EE | MOVL | CALL ACMODE, R3 | |
| | 00000000G | 00 | 16 | 000F5 | JSB | EXESPROBER | |
| 61 | | 50 | E9 | 000FB | BLBC | RO, 17\$ | |
| 03 | | 58 | E8 | 000FE | BLBS | ACL_STATUS, 11\$ | 2539 |
| | | 00B1 | 31 | 00101 | BRW | 20\$ | |
| 7E | | 57 | 7D | 00104 | MOVQ | SIZE, -(SP) | 2540 |
| | 00000000' | EF | 9F | 00107 | PUSHAB | ACL_CONTEXT | |
| | 00000000' | EF | DD | 0010D | PUSHL | ACL_QUEUE_HEAD | |
| 00000000G | 00 | 04 | FB | 00113 | CALLS | #4, ACL_M0DENTRY | |
| | | 7F | 11 | 0011A | BRB | 22\$ | |
| 50 | | 58 | D0 | 0011C | MOVL | BUFFER, RO | 2545 |
| 51 | | 57 | D0 | 0011F | MOVL | SIZE, R1 | |
| 53 | 00000000' | EF | D0 | 00122 | MOVL | CALL ACMODE, R3 | |
| | 00000000G | 00 | 16 | 00129 | JSB | EXESPROBEW | |
| 7E | | 50 | E9 | 0012F | BLBC | RO, 24\$ | |
| | | 7E | D4 | 00132 | CLRL | -(SP) | 2547 |
| 7E | | 57 | 7D | 00134 | MOVQ | SIZE, -(SP) | |
| | 00000000' | EF | 9F | 00137 | PUSHAB | ACL_CONTEXT | |
| | 00000000' | EF | DD | 0013D | PUSHL | ACL_QUEUE_HEAD | |
| 00000000G | 00 | 05 | FB | 00143 | CALLS | #5, ACL_FINDENTRY | |
| | | 7D | 11 | 0014A | BRB | 25\$ | |
| 50 | | 58 | D0 | 0014C | MOVL | BUFFER, RO | 2552 |
| 51 | | 57 | D0 | 0014F | MOVL | SIZE, R1 | |
| 53 | 00000000' | EF | D0 | 00152 | MOVL | CALL ACMODE, R3 | |
| | 00000000G | 00 | 16 | 00159 | JSB | EXESPROBEW | |
| 7C | | 50 | E9 | 0015F | BLBC | RO, 27\$ | |
| | | 7E | D4 | 00162 | CLRL | -(SP) | 2554 |
| 7E | | 57 | 7D | 00164 | MOVQ | SIZE, -(SP) | |
| | 00000000' | EF | 9F | 00167 | PUSHAB | ACL_CONTEXT | |
| | 00000000' | EF | DD | 0016D | PUSHL | ACL_QUEUE_HEAD | |
| 00000000G | 00 | 05 | FB | 00173 | CALLS | #5, ACL_FINDTYPE | |
| | | 7B | 11 | 0017A | BRB | 28\$ | |
| 03 | | 6E | E9 | 0017C | BLBC | (SP), 19\$ | 2559 |
| | | 0114 | 31 | 0017F | BRW | 40\$ | |
| 03 | | 58 | E8 | 00182 | BLBS | ACL_STATUS, 21\$ | 2561 |
| | | 0146 | 31 | 00185 | BRW | 45\$ | |
| | 00000000' | EF | 9F | 00188 | PUSHAB | ACL_CONTEXT | 2562 |
| | 00000000' | EF | DD | 0018E | PUSHL | ACL_QUEUE_HEAD | |
| 00000000G | 00 | 02 | FB | 00194 | CALLS | #2, ACL_DELETEACL | |
| | | 5A | 11 | 0019B | BRB | 28\$ | |
| 50 | | 58 | D0 | 0019D | MOVL | BUFFER, RO | 2567 |
| 51 | | 57 | D0 | 001A0 | MOVL | SIZE, R1 | |
| 53 | 00000000' | EF | D0 | 001A3 | MOVL | CALL ACMODE, R3 | |
| | 00000000G | 00 | 16 | 001AA | JSB | EXESPROBEW | |

| | | | | | | | | |
|-----------|-----------|----|-------|-------|--------|--------------------|--------------------------------------|------|
| 59 | 50 | E9 | 00180 | 24\$: | BLBC | R0, 30\$ | | |
| 7E | 57 | 7D | 00183 | | MOVQ | SIZE, -(SP) | 2569 | |
| | EF | 9F | 00186 | | PUSHAB | ACL_CONTEXT | | |
| 00000000G | 00 | EF | DD | 0018C | PUSHL | ACL_QUEUE_HEAD | | |
| | 04 | FB | 001C2 | | CALLS | #4, ACL_READACL | | |
| | 5A | 11 | 001C9 | 25\$: | BRB | 31\$ | | |
| 50 | 58 | DD | 001CB | 26\$: | MOVL | BUFFER, R0 | 2574 | |
| 51 | 57 | DD | 001CE | | MOVL | SIZE, R1 | | |
| 53 | 00000000' | EF | DD | 001D1 | MOVL | CALL_ACMODE, R3 | | |
| | 00000000G | 00 | 16 | 001D8 | JSB | EXESPROBEW | | |
| 61 | 50 | E9 | 001DE | 27\$: | BLBC | R0, 33\$ | | |
| 7E | 57 | 7D | 001E1 | | MOVQ | SIZE, -(SP) | 2576 | |
| | EF | 9F | 001E4 | | PUSHAB | ACL_CONTEXT | | |
| 00000000G | 00 | EF | DD | 001EA | PUSHL | ACL_QUEUE_HEAD | | |
| | 04 | FB | 001F0 | | CALLS | #4, ACL_ACLLENGTH | | |
| | 2C | 11 | 001F7 | 28\$: | BRB | 31\$ | | |
| 50 | 58 | DD | 001F9 | 29\$: | MOVL | BUFFER, R0 | 2581 | |
| 51 | 57 | DD | 001FC | | MOVL | SIZE, R1 | | |
| 53 | 00000000' | EF | DD | 001FF | MOVL | CALL_ACMODE, R3 | | |
| | 00000000G | 00 | 16 | 00206 | JSB | EXESPROBEW | | |
| 33 | 50 | E9 | 0020C | 30\$: | BLBC | R0, 33\$ | | |
| 7E | 57 | 7D | 0020F | | MOVQ | SIZE, -(SP) | 2583 | |
| | EF | 9F | 00212 | | PUSHAB | ACL_CONTEXT | | |
| 00000000G | 00 | EF | DD | 00218 | PUSHL | ACL_QUEUE_HEAD | | |
| | 04 | FB | 0021E | | CALLS | #4, ACL_READACE | | |
| 5B | 50 | DD | 00225 | 31\$: | MOVL | R0, ACL_STATUS | | |
| | 6F | 11 | 00228 | | BRB | 41\$ | 2510 | |
| 04 | 57 | D1 | 0022A | 32\$: | CMPL | SIZE, #4 | 2590 | |
| | 67 | 1F | 0022D | | BLSSU | 40\$ | | |
| 50 | 58 | DD | 0022F | | MOVL | BUFFER, R0 | 2592 | |
| 51 | 57 | DD | 00232 | | MOVL | SIZE, R1 | | |
| 53 | 00000000' | EF | DD | 00235 | MOVL | CALL_ACMODE, R3 | | |
| | 00000000G | 00 | 16 | 0023C | JSB | EXESPROBEW | | |
| 6C | 50 | E9 | 00242 | 33\$: | BLBC | R0, 43\$ | | |
| 7E | 03 | 7D | 00245 | | MOVQ | #3, -(SP) | 2606 | |
| | 7E | 7C | 00248 | | CLRQ | -(SP) | | |
| | 7E | D4 | 0024A | | CLRL | -(SP) | | |
| 03 | 00000000' | EF | D1 | 0024C | CMPL | CALL_ACMODE, #3 | | |
| | 00000000' | 08 | 12 | 00253 | BNEQ | 34\$ | | |
| | | EF | DD | 00255 | PUSHL | PARENT_ID | | |
| | | 02 | 11 | 0025B | BRB | 35\$ | | |
| | | 7E | D4 | 0025D | CLRL | -(SP) | | |
| | 00000000' | EF | 9F | 0025F | 34\$: | PUSHAB | LOCK_RESNAM | |
| | | 1C | DD | 00265 | 35\$: | PUSHL | #28 | |
| 0A | 28 | AE | 9F | 00267 | PUSHAB | LOCAL_IOSB | | |
| | | 5A | D1 | 0026A | CMPL | FUNCTION_CODE, #10 | | |
| | | 04 | 12 | 0026D | BNEQ | 36\$ | | |
| | | 01 | DD | 0026F | PUSHL | #1 | | |
| | | 02 | 11 | 00271 | BRB | 37\$ | | |
| | | 04 | DD | 00273 | 36\$: | PUSHL | #4 | |
| | | 7E | D4 | 00275 | 37\$: | CLRL | -(SP) | |
| 00000000G | 00 | 08 | FB | 00277 | CALLS | #11, SYSSENQ | | |
| | 59 | 50 | DD | 0027E | MOVL | R0, STATUS | | |
| | 04 | 59 | E9 | 00281 | BLBC | STATUS, 38\$ | 2607 | |
| | 59 | 08 | AE | 3C | 00284 | MOVZWL | LOCAL_IOSB, STATUS | |
| 57 | 00 | 0C | AE | 04 | 2C | 00288 | 38\$: | |
| | | | | 68 | 0028E | MOVCS | #4, LOCAL_IOSB+4, #0, SIZE, (BUFFER) | 2610 |

| | | | | | | | | | |
|----|--------------|----|-----------|-------|-------|-------|--------|--------------------------------------|------|
| | | | 3D | 11 | 0028F | BRB | 45\$ | 2510 | |
| | | 04 | 57 | D1 | 00291 | 39\$: | CMP | SIZE, #4 | 2617 |
| | | | 05 | 1E | 00294 | BGEQU | 42\$ | | |
| | | 59 | 14 | D0 | 00296 | 40\$: | MOVL | #20, STATUS | 2618 |
| | | | 33 | 11 | 00299 | 41\$: | BRB | 45\$ | |
| | | 50 | 58 | D0 | 0029B | 42\$: | MOVL | BUFFER, R0 | 2619 |
| | | 51 | 57 | D0 | 0029E | | MOVL | SIZE, R1 | |
| | | 53 | EF | D0 | 002A1 | | MOVL | CALL ACMODE, R3 | |
| | | | 00 | 16 | 002AB | | JSB | EXESPROBER | |
| | | 05 | 50 | E8 | 002AE | | BLBS | R0, 44\$ | |
| | | 59 | 0C | D0 | 002B1 | 43\$: | MOVL | #12, STATUS | 2620 |
| | | | 18 | 11 | 002B4 | | BRB | 45\$ | |
| 04 | 00 | 68 | 57 | 2C | 002B6 | 44\$: | MOVC5 | SIZE, (BUFFER), #0, #4, LOCAL_LOCKID | 2623 |
| | | | AE | | 002BB | | | | |
| | | | 7E | 7C | 002BD | | CLRQ | -(SP) | 2624 |
| | | | 7E | D4 | 002BF | | CLRL | -(SP) | |
| | | | AE | DD | 002C1 | | PUSHL | LOCAL_LOCKID | |
| | 00000000G | 00 | 04 | FB | 002C4 | | CALLS | #4, SYSSDEQ | |
| | | 59 | 50 | D0 | 002CB | | MOVL | R0, STATUS | |
| | | 10 | 59 | E9 | 002CE | 45\$: | BLBC | STATUS, 49\$ | 2635 |
| 02 | | 56 | AC | F2 | 002D1 | 46\$: | A0BLSS | ITEM_COUNT, J, 47\$ | 2500 |
| | | | 03 | 11 | 002D6 | | BRB | 48\$ | |
| | | | FD65 | 31 | 002D8 | 47\$: | BRW | 3\$ | |
| | | 03 | 59 | E9 | 002DB | 48\$: | BLBC | STATUS, 49\$ | 2642 |
| | | 59 | 5B | D0 | 002DE | | MOVL | ACL_STATUS, STATUS | |
| | 50 00000000' | EF | 24 | C3 | 002E1 | 49\$: | SUBL3 | #36, ACL_QUEUE_HEAD, R0 | 2646 |
| | | 54 | 00000000G | 00 | D0 | 002E9 | MOVL | CTL\$GL_PCB, R4 | |
| | | | 00000000G | 00 | 16 | 002F0 | JSB | SCH\$UNLOCK | |
| | | 50 | 59 | D0 | 002F6 | | MOVL | STATUS, R0 | 2648 |
| | | | 04 | 002F9 | | RET | | | 2650 |

; Routine Size: 762 bytes, Routine Base: \$CODE\$ + 1A87


```
2667 2651 1 %SBTTL 'RUNDOWN CHANGE_ACL - run down $CHANGE_ACL context'
2668 2652 1 GLOBAL ROUTINE RUNDOWN_CHANGE_ACL =
2669 2653 1
2670 2654 1 ++
2671 2655 1
2672 2656 1 FUNCTIONAL DESCRIPTION:
2673 2657 1
2674 2658 1 This routine is called to perform the appropriate ACL operations.
2675 2659 1 The code is checked for validity and, when necessary, the buffer
2676 2660 1 is probed for the desired access.
2677 2661 1
2678 2662 1 CALLING SEQUENCE:
2679 2663 1 RUNDOWN_CHANGE_ACL ( )
2680 2664 1
2681 2665 1 INPUT PARAMETERS:
2682 2666 1 NONE
2683 2667 1
2684 2668 1 IMPLICIT INPUTS:
2685 2669 1 PARENT_ID: lock ID of parent for ACL locks
2686 2670 1
2687 2671 1 OUTPUT PARAMETERS:
2688 2672 1 NONE
2689 2673 1
2690 2674 1 IMPLICIT OUTPUTS:
2691 2675 1 NONE
2692 2676 1
2693 2677 1 ROUTINE VALUE:
2694 2678 1 1
2695 2679 1
2696 2680 1 SIDE EFFECTS:
2697 2681 1 All ACL locks taken out by user mode $CHANGE_ACL calls, plus the
2698 2682 1 parent lock, are dequeued.
2699 2683 1
2700 2684 1 --
2701 2685 1
2702 2686 2 BEGIN
2703 2687 2
2704 2688 2 IF .PARENT_ID NEQ 0
2705 2689 2 THEN
2706 2690 2 BEGIN
2707 2691 2 $DEQ (LKID = .PARENT_ID,
2708 2692 2 FLAGS = LCKSM DEQALL);
2709 2693 2 $DEQ (LKID = .PARENT_ID);
2710 2694 2 PARENT_ID = 0;
2711 2695 2 END;
2712 2696 2
2713 2697 1
2714 2698 1 END;

! End of routine RUNDOWN_CHANGE_ACL
```

```
53 00000000G 00 000C 00000
52 00000000' EF 9E 00002
50 62 D0 00010
```

```
.ENTRY RUNDOWN_CHANGE_ACL, Save R2,R3
MOVAB SYSSDEQ, R3
MOVAB PARENT_ID, R2
MOVL PARENT_ID, R0
```

```
: 2652
:
: 2688
```

63

63

50

| | | |
|----|-------|------------|
| 14 | 13 | 00013 |
| 01 | DD | 00015 |
| 7E | 7C | 00017 |
| 50 | DD | 00019 |
| 04 | FB | 0001B |
| 7E | 7C | 0001E |
| 7E | D4 | 00020 |
| 62 | DD | 00022 |
| 04 | FB | 00024 |
| 62 | D4 | 00027 |
| 01 | DD | 00029 1\$: |
| 04 | 0002C | |

| | |
|-------|-------------|
| BEQL | 1\$ |
| PUSHL | #1 |
| CLRQ | -(SP) |
| PUSHL | R0 |
| CALLS | #4, SYSSDEQ |
| CLRQ | -(SP) |
| CLRL | -(SP) |
| PUSHL | PARENT_ID |
| CALLS | #4, SYSSDEQ |
| CLRL | PARENT_ID |
| MOVL | #1, R0 |
| RET | |

2692

2693

2694

2698

; Routine Size: 45 bytes, Routine Base: \$CODE\$ + 1D81

| | | |
|--------|------|----------|
| : 2715 | 2699 | 1 |
| : 2716 | 2700 | 1 END |
| : 2717 | 2701 | 0 ELUDOM |

PSECT SUMMARY

| Name | Bytes | Attributes |
|--------------|-------|--|
| \$OWNS | 1171 | NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2) |
| \$PLITS | 1396 | NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2) |
| _LIB\$KEYOS | 42 | NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1) |
| _LIB\$STATES | 664 | NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1) |
| _LIB\$KEY1\$ | 213 | NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1) |
| \$CODE\$ | 7598 | NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2) |
| . ABS . | 0 | NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0) |

Library Statistics

| File | Total | Symbols Loaded | Percent | Pages Mapped | Processing Time |
|-----------------------------------|-------|----------------|---------|--------------|-----------------|
| \$255\$DUA28:[SYSLIB]LIB.L32;1 | 18619 | 211 | 1 | 1000 | 00:01.8 |
| \$255\$DUA28:[SYSLIB]TPAMAC.L32;1 | 42 | 28 | 66 | 14 | 00:00.2 |

| | |
|----------------|---|
| : Information: | 1 |
| : Warnings: | 0 |
| : Errors: | 0 |

SYSACLSRV
V04-000

RUNDOWN_CHANGE_ACL - run down \$CHANGE_ACL conte

D 5
16-Sep-1984 01:51:51
14-Sep-1984 12:40:53

VAX-11 Bliss-32 V4.0-742
[LOADSS.SRC]SYSACLSRV.B32;1

Page 111
(17)

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SYSACLSRV/OBJ=OBJ\$:SYSACLSRV MSRC\$:SYSACLSRV/UPDATE=(ENH\$:SYSACLSRV)

: Size: 7598 code + 3486 data bytes
: Run Time: 03:19.4
: Elapsed Time: 06:02.3
: Lines/CPU Min: 812
: Lexemes/CPU-Min: 37990
: Memory Used: 1462 pages
: Compilation Complete

0220

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

0221 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

